明 細 書

認証システム及び遠隔分散保存システム

技術分野

本発明は、認証に関する情報の漏洩に強い認証システムとこの認証システムを用いて安全にデータを保存することができる遠隔分散保存システムに関する。

本願は、２００３年１０月２８日に出願された特願２００３－３６７５２７号に対し優先権を主張し、その内容をここに援用する。

背景技術

従来からユーザの端末装置とサーバとの間において認証を行う方法として、ユーザＩＤとユーザのみが知っているパスワードとを端末装置から入力し、サーバ側に蓄えられている情報と一致すれば、正当なユーザであることを認証する方法が知られている。

しかし、この方法は、端末装置とサーバ間の通信経路中において、不正にこれらの情報が盗まれてしまうと、簡単に不正利用を許してしまうため、ＳＳＬ（文献１）、ＴＬＳ（文献２）、ＳＳＨ（文献３）等の暗号化技術を用いて、情報の送受信を行うのが一般的である。これは、パスワードと秘密の値と公開されている値とを使用して認証を行うものである。

（文献１）A.Frier,P.Karlton,and P.Kocher.The SSL 3.0 Protocol.Netscape Communications Corp.,1996,http://wp.netscape.com/eng/ss13/

（文献２）IETF(Internet Engineering Task Force).Transport Layer security (tls) Charter.http://www.ietf.org/html.charters/tls-charter.html

（文献３）IETF(Internet Engineering Task Force).Secure Shell(secsh)Charter.http://www.ietf.org/html.charters/secsh-charter.html

しかしながら、非特許文献１～３に示す方法は、ユーザの端末装置側からパスワードで暗号化された情報あるいはサーバ側からパスワード認証データが漏れた場合は、オフラインの解析作業によってパスワードを求めることができてしまうという問題がある。サーバに対してオンラインでパスワード入力を繰り返し行う方法は、パスワードを間違えた回数に応じてアクセスを拒否するなどの対策を講じることが可能であるが、オフラインの解析作業は、防止策を講じることができないという問題もある。また、パスワードが漏洩してしまった場合、このパスワードでログインできるシステム内に保存されているデータも漏洩してしまうという問題もある。

発明の開示

本発明の目的は、情報の漏洩に強く、安全に暗号鍵の交換を行うことができる認証システムを提供することである。

本発明の目的は、本発明による認証システムを用いて安全にデータを保存することができる遠隔分散保存システムを提供することである。

本発明の要旨は、端末装置とサーバ間において相互に認証を行う認証システムであって、前記端末装置は、ユーザが予め決定しておいたパスワードに基づいて、サーバ登録用のパスワード認証データＨとユーザ保存用の認証情報Ｐ’を求めるデータ伸長手段と、前記データ伸長手段によって求めた認証情報Ｐ’を予め記憶しておく記憶手段と、前記記憶手段から読み出した認証情報Ｐ’と認証時に入力されたパスワードを入力として所定の計算式により値Ｐを求める結合手段と、前記値Ｐと内部において発生させた乱数を入力として所定の計算式により値Ｙ１を求め、前記サーバへ送信するマスク演算手段と、前記値Ｐと内部において発生させた乱数と前記サーバから受信した値Ｙ２を入力として所定の計算式により値ＭＫを求めるマスター鍵生成手段と、前記値ＭＫを入力として所定の計算式により値Ｖ１を求め、サーバへ送信するとともに、前記サーバから受信した値Ｖ２と前記値ＭＫを入力として所定の計算式による値Ｖ２と比較照合し、一致した場合にサーバを認証する認証結果判断手段とを備

え、前記サーバは、前記データ伸長手段によって求めたパスワード認証データＨを予め記憶しておく記憶手段と、前記記憶手段から読み出したパスワード認証データＨと内部において発生させた乱数を入力として所定の計算式により値Ｙ２を求め、前記端末装置へ送信するマスク演算手段と、前記パスワード認証データＨと内部において発生させた乱数と前記端末装置から受信した値Ｙ１を入力として所定の計算式により値ＭＫを求めるマスター鍵生成手段と、前記値ＭＫを入力として所定の計算式により値Ｖ２を求め、端末装置へ送信するとともに、前記端末装置から受信した値Ｖ１と前記値ＭＫを入力として所定の計算式による値Ｖ１と比較照合し、一致した場合に端末装置を認証する認証結果判断手段とを備える。

本発明の要旨は、端末装置とサーバ間において相互に認証を行う認証システムにおける端末装置上で動作する認証プログラムであって、ユーザが予め決定しておいたパスワードに基づいて、サーバ登録用のパスワード認証データＨとユーザ保存用の認証情報Ｐ’を求めるデータ伸長処理と、前記データ伸長処理によって求めた認証情報Ｐ’を予め記憶しておく記憶処理と、前記記憶処理により記憶しておいた認証情報Ｐ’と認証時に入力されたパスワードを入力として所定の計算式により値Ｐを求める結合処理と、前記値Ｐと内部において発生させた乱数を入力として所定の計算式により値Ｙ１を求め、前記サーバへ送信するマスク演算処理と、前記値Ｐと内部において発生させた乱数と前記サーバから受信した値Ｙ２を入力として所定の計算式により値ＭＫを求めるマスター鍵生成処理と、前記値ＭＫを入力として所定の計算式により値Ｖ１を求め、サーバへ送信するとともに、前記サーバから受信した値Ｖ２と前記値ＭＫを入力として所定の計算式による値Ｖ２と比較照合し、一致した場合にサーバを認証する認証結果判断処理とをコンピュータに行わせる。

本発明の要旨は、端末装置とサーバ間において相互に認証を行う認証システムにおけるサーバ上で動作する認証プログラムであって、パスワード認証データＨを予め記憶しておく記憶処理と、前記記憶処理により記憶しておいたパス

ワード認証データＨと内部において発生させた乱数を入力として所定の計算式により値Ｙ２を求め、前記端末装置へ送信するマスク演算処理と、前記パスワード認証データＨと内部において発生させた乱数と前記端末装置から受信した値Ｙ１を入力として所定の計算式により値ＭＫを求めるマスター鍵生成処理と、前記値ＭＫを入力として所定の計算式により値Ｖ２を求め、端末装置へ送信するとともに、前記端末装置から受信した値Ｖ１と前記値ＭＫを入力として所定の計算式による値Ｖ１と比較照合し、一致した場合に端末装置を認証する認証結果判断処理とをコンピュータに行わせる。

本発明の要旨は、端末装置とサーバ間において相互に認証を行う認証システムであって、前記端末装置は、ユーザが予め決定しておいたパスワードに基づいて、サーバ登録用のパスワード認証データＨとユーザ保存用の認証情報Ｐ’を求めるデータ伸長手段と、前記データ伸長手段によって求めた認証情報Ｐ’とＲＳＡ鍵生成手段によって求めたＲＳＡ公開鍵（Ｎ，ｅ）を予め記憶しておく記憶手段と、前記記憶手段から読み出した認証情報Ｐ’と認証時に入力されたパスワードを入力として所定の計算式により値Ｗを求める結合手段と、前記値Ｗと前記記憶手段から読み出したＲＳＡ公開鍵（Ｎ，ｅ）と内部において発生させた乱数Ｔを入力として所定の計算式により値Ｚを求め、前記サーバへ送信するマスク演算手段と、前記サーバから受信した値Ｖ２と前記乱数Ｔを入力として所定の計算式による値Ｖ２と比較照合し、一致した場合にサーバを認証する認証結果判断手段と、前記乱数Ｔを入力として所定の計算式により値Ｖ１を求め、前記サーバへ送信する検証子生成手段とを備え、前記サーバは、ＲＳＡ公開鍵（Ｎ，ｅ）とＲＳＡ秘密鍵（Ｎ，ｄ）を求めるＲＳＡ鍵生成手段と、前記ＲＳＡ鍵生成手段によって求めたＲＳＡ秘密鍵（Ｎ，ｄ）と前記データ伸長手段によって求めたパスワード認証データＨを予め記憶しておく記憶手段と、前記記憶手段から読み出したＲＳＡ秘密鍵（Ｎ，ｄ）とパスワード認証データＨと前記端末装置から受信した値Ｚを入力として所定の計算式により値Ｔを求めるマスター鍵生成手段と、前記値Ｔを入力として所定の計算式により値Ｖ２を求め、前記端末装置へ送信する検証子生成手段と、前記端末装置から

受信した値Ｖ１と前記値Ｔを入力として所定の計算式による値Ｖ１と比較照合し、一致した場合に端末装置を認証する認証結果判断手段とを備える。

本発明の要旨は、端末装置とサーバ間において相互に認証を行う認証システムにおける端末装置上で動作する認証プログラムであって、ユーザが予め決定しておいたパスワードに基づいて、サーバ登録用のパスワード認証データＨとユーザ保存用の認証情報Ｐ’を求めるデータ伸長処理と、前記データ伸長処理によって求めた認証情報Ｐ’とＲＳＡ鍵生成処理によって求めたＲＳＡ公開鍵（Ｎ，ｅ）を予め記憶しておく記憶処理と、前記記憶処理により記憶しておいた認証情報Ｐ’と認証時に入力されたパスワードを入力として所定の計算式により値Ｗを求める結合処理と、前記値Ｗと前記記憶処理により記憶しておいたＲＳＡ公開鍵（Ｎ，ｅ）と内部において発生させた乱数Ｔを入力として所定の計算式により値Ｚを求め、前記サーバへ送信するマスク演算処理と、前記サーバから受信した値Ｖ２と前記乱数Ｔを入力として所定の計算式による値Ｖ２と比較照合し、一致した場合にサーバを認証する認証結果判断処理と、前記乱数Ｔを入力として所定の計算式により値Ｖ１を求め、前記サーバへ送信する検証子生成処理とをコンピュータに行わせる。

本発明の要旨は、端末装置とサーバ間において相互に認証を行う認証システムにおけるサーバ上で動作する認証プログラムであって、ＲＳＡ公開鍵（Ｎ，ｅ）とＲＳＡ秘密鍵（Ｎ，ｄ）を求めるＲＳＡ鍵生成処理と、前記ＲＳＡ鍵生成処理によって求めたＲＳＡ秘密鍵（Ｎ，ｄ）とパスワード認証データＨを予め記憶しておく記憶処理と、前記記憶処理により記憶しておいたＲＳＡ秘密鍵（Ｎ，ｄ）とパスワード認証データＨと前記端末装置から受信した値Ｚを入力として所定の計算式により値Ｔを求めるマスター鍵生成処理と、前記値Ｔを入力として所定の計算式により値Ｖ２を求め、前記端末装置へ送信する検証子生成処理と、前記端末装置から受信した値Ｖ１と前記値Ｔを入力として所定の計算式による値Ｖ１と比較照合し、一致した場合に端末装置を認証する認証結果判断処理とをコンピュータに行わせる。

本発明の要旨は、端末装置と複数のサーバ間において相互に認証を行い、前記端末装置内の保存対象のデータを前記サーバ内に分散して保存する遠隔分散保存システムであって、前記端末装置は、ユーザが予め決定しておいたパスワードに基づいて、サーバ登録用のパスワード認証データＨとユーザ保存用の認証情報Ｐ’を求めるデータ伸長手段と、前記データ伸長手段によって求めた認証情報Ｐ’を予め記憶しておく記憶手段と、前記記憶手段から読み出した認証情報Ｐ’と認証時に入力されたパスワードを入力として所定の計算式により値Ｐを求める結合手段と、前記値Ｐと内部において発生させた乱数を入力として所定の計算式により値Ｙ１を求め、前記サーバへ送信するマスク演算手段と、前記値Ｐと内部において発生させた乱数と前記サーバから受信した値Ｙ２を入力として所定の計算式により値ＭＫを求めるマスター鍵生成手段と、前記値ＭＫを入力として所定の計算式により値Ｖ１を求め、サーバへ送信するとともに、前記サーバから受信した値Ｖ２と値Ｖ１と比較照合し、一致した場合にサーバを認証する認証結果判断手段と、サーバの認証が行われた場合に、セッション鍵ＳＫをサーバの数だけ生成するセッション鍵生成手段と、前記保存対象のデータを分割して、認証したサーバの数と同数の分割データを得るデータ分割手段と、前記分割データのそれぞれと保存対象のデータを識別する識別情報とを、保存先のサーバと共有した前記セッション鍵ＳＫを用いて暗号化して、各サーバに対して送信するデータ保存手段と、分割データが保存された各サーバから分割データを受信し、前記保存対象のデータを復元するデータ復元手段とを備え、前記サーバは、前記データ伸長手段によって求めたパスワード認証データＨを予め記憶しておく記憶手段と、前記記憶手段から読み出したパスワード認証データＨと内部において発生させた乱数を入力として所定の計算式により値Ｙ２を求め、前記端末装置へ送信するマスク演算手段と、前記パスワード認証データＨと内部において発生させた乱数と前記端末装置から受信した値Ｙ１を入力として所定の計算式により値ＭＫを求めるマスター鍵生成手段と、前記値ＭＫを入力として所定の計算式により値Ｖ２を求め、端末装置へ送信するとともに、前記端末装置から受信した値Ｖ１と値Ｖ２と比較照合し、

一致した場合に端末装置を認証する認証結果判断手段と、端末装置の認証が行われた場合に、セッション鍵を生成するセッション鍵生成手段と、端末装置から受信した分割データを受信するデータ受信手段と、前記分割データを記憶するデータ記憶手段と、前記データ記憶手段に保存されている分割データを読み出して端末装置へ送信するデータ送信手段とを備えたことを特徴とする遠隔分散保存システム。

本発明の要旨は、端末装置と複数のサーバ間において相互に認証を行い、前記端末装置内の保存対象のデータを前記サーバ内に分散して保存する遠隔分散保存システムにおける端末装置上で動作する遠隔分散保存プログラムであって、ユーザが予め決定しておいたパスワードに基づいて、サーバ登録用のパスワード認証データＨとユーザ保存用の認証情報Ｐ’を求めるデータ伸長処理と、前記データ伸長処理によって求めた認証情報Ｐ’を予め記憶しておく記憶処理と、前記記憶処理から読み出した認証情報Ｐ’と認証時に入力されたパスワードを入力として所定の計算式により値Ｐを求める結合処理と、前記値Ｐと内部において発生させた乱数を入力として所定の計算式により値Ｙ１を求め、前記サーバへ送信するマスク演算処理と、前記値Ｐと内部において発生させた乱数と前記サーバから受信した値Ｙ２を入力として所定の計算式により値ＭＫを求めるマスター鍵生成処理と、前記値ＭＫを入力として所定の計算式により値Ｖ１を求め、サーバへ送信するとともに、前記サーバから受信した値Ｖ２と値Ｖ１と比較照合し、一致した場合にサーバを認証する認証結果判断処理と、サーバの認証が行われた場合に、セッション鍵ＳＫをサーバの数だけ生成するセッション鍵生成処理と、前記保存対象のデータを分割して、認証したサーバの数と同数の分割データを得るデータ分割処理と、前記分割データのそれぞれと保存対象のデータを識別する識別情報とを、保存先のサーバと共有した前記セッション鍵ＳＫを用いて暗号化して、各サーバに対して送信するデータ保存処理と、分割データが保存された各サーバから分割データを受信し、前記保存対象のデータを復元するデータ復元処理とをコンピュータに行わせることを特徴とする遠隔分散保存プログラム

本発明の要旨は、端末装置と複数のサーバ間において相互に認証を行い、前記端末装置内の保存対象のデータを前記サーバ内に分散して保存する遠隔分散保存システムにおけるサーバ上で動作する遠隔分散保存プログラムであって、データ伸長処理によって求めたパスワード認証データＨを予め記憶しておく記憶処理と、前記記憶処理から読み出したパスワード認証データＨと内部において発生させた乱数を入力として所定の計算式により値Ｙ２を求め、前記端末装置へ送信するマスク演算処理と、前記パスワード認証データＨと内部において発生させた乱数と前記端末装置から受信した値Ｙ１を入力として所定の計算式により値ＭＫを求めるマスター鍵生成処理と、前記値ＭＫを入力として所定の計算式により値Ｖ２を求め、端末装置へ送信するとともに、前記端末装置から受信した値Ｖ１と値Ｖ２と比較照合し、一致した場合に端末装置を認証する認証結果判断処理と、端末装置の認証が行われた場合に、セッション鍵を生成するセッション鍵生成処理と、端末装置から受信した分割データを受信するデータ受信処理と、前記分割データを記憶するデータ記憶処理と、前記データ記憶処理に保存されている分割データを読み出して端末装置へ送信するデータ送信処理とをコンピュータに行わせることを特徴とする遠隔分散保存プログラム。

図面の簡単な説明

図１は、本発明の一実施形態における端末装置の構成を示すブロック図である。

図２は、図１に示すデータ伸長器１１の構成を示すブロック図である。

図３は、図１に示すデータ伸長器１１の構成を示すブロック図である。

図４は、図１に示すデータ伸長器１１の構成を示すブロック図である。

図５は、相互認証及び鍵交換を行う装置の構成を示すブロック図である。

図６は、図５に示す端末装置１の構成を示すブロック図である。

図７は、図５に示すサーバ２の構成を示すブロック図である。

図８は、図１に示すデータ伸長器１１の構成を示すブロック図である。

図９は、パスワード認証データ更新－１の端末装置１の構成を示すブロック図である。

図１０は、パスワード認証データ更新－１のサーバ２の構成を示すブロック図である。

図１１は、パスワード認証データ更新－２の端末装置１の構成を示すブロック図である。

図１２は、パスワード認証データ更新－２のサーバ２の構成を示すブロック図である。

図１３は、図１に示すデータ伸長器１１の構成を示すブロック図である。

図１４は、図１に示すデータ伸長器１１の構成を示すブロック図である。

図１５は、安全な通信を利用して初期化処理を行う場合のサーバ２の構成を示すブロック図である。

図１６は、安全ではない通信を利用して初期化処理を行う場合の端末装置１及びサーバ２の構成を示すブロック図である。

図１７は、図５に示す端末装置１の構成を示すブロック図である。

図１８は、図５に示すサーバ２の構成を示すブロック図である。

図１９は、マスター鍵を利用して更新化処理を行う場合の端末装置１の構成を示すブロック図である。

図２０は、マスター鍵を利用して更新化処理を行う場合のサーバ２の構成を示すブロック図である。

図２１は、端末に分散データを保存しない場合の遠隔分散保存装置５の構成を示すブロック図である。

図２２は、図２１に示すデータ分散器５１の構成を示すブロック図である。

図２３は、端末にデータを保存しない場合の遠隔分散保存装置５の構成を示すブロック図である。

図２４は、図２３に示すデータ復元器５４の構成を示すブロック図である。

図２５は、端末にも分散データを保存する場合の遠隔分散保存装置５の構成を示すブロック図である。

図２６は、図２５に示すデータ分割器５１の構成を示すブロック図である。

図２７は、端末にも分散データを保存する場合の遠隔分散保存装置５の構成を示すブロック図である。

図２８は、図２７に示すデータ復元器５４の構成を示すブロック図である。

発明を実施するための最良の形態

＜第１実施例＞

以下、図面を参照しつつ、本発明の好適な実施例について説明する。ただし、本発明は以下の各実施例に限定されるものではなく、例えばこれら実施例の構成要素同士を適宜組み合わせてもよい。

この認証システムは、ユーザの端末装置とサーバの認証装置がお互いに相互認証しながら同じセッション鍵を確保するためのシステムである。

ここで、以下の説明において用いる記号について説明しておく。

ｐ，ｑは素数であり、ｑ｜ｐ－１という関係がある。ｑ｜ｐ－１は、ｑはｐ－１を割りきることのできる値であることを意味する。また、ｇ，ｈはｍｏｄｐ上の位数ｑの有限体(群)Ｇ＝｛g^j ｍｏｄ ｐ：０≦ｊ＜ｑ｝の生成元である（楕円曲線上の群でも同じように構成できる）。ここで、“g^j ｍｏｄ ｐ”は、法指数演算で、ｇをｊ乗した値をｐで割った残り（Ｒｅｍａｉｎｄｅｒ）という意味である。ここで、ｇは($1<g<p-1$，$g^q \equiv 1 \bmod p$，$g^j \not\equiv 1 \bmod p$（$0<j<q$））であり、ｈは$h=g^a \bmod p$である。つまり、ｐ，ｑは演算体系（素体の標数）を示す。例えば、$H=h^x \bmod p$（$0<x<q$）でｘは秘密情報である（つまり、Ｈが与えられた時、$x=\log_h H$を求めるのは数学的に難しい問題；Ｈの生成元ｈに対する離散対数問題）。また、乱数発生器から発生される乱数は$R \in (Z/qZ)^*$を無作為に生成する。ここで、$(Z/qZ)^*$は｛１，２，・・・，ｑ｝の集合を示す。また、Ｎはパスワードの長さを示す。また、‖は値を連結（ｃｏｎｃａｔｅｎａｔｉｏｎ）するという意味である。

＜端末装置の初期化＞

ユーザは、サーバに対して個人登録したい時、自分の端末装置の初期化を行

う。図１は、ユーザの端末装置の初期化処理の構成を示すブロック図である。初期化は、ユーザがパスワードを入力すると、データ伸張器１１によって、サーバ登録用のパスワード認証データＨと、ユーザ保存用の値Ｐ’が生成され、パスワード認証データＨは、サーバに受け渡され、値Ｐ’は、メモリ１２へ保存する。ここで、データ伸長器１１は、多項式、多項式とハッシュ関数、ハッシュ関数、擬似乱数発生器などで構成することが可能である。

（１）多項式を利用する場合（その１）

初めに、図２を参照して、多項式を利用する場合（その１）について説明する。

まず、多項式発生器１１１によりランダムに多項式を発生する。このとき、登録するサーバの数が一つだったらｘを変数とする１次多項式（ｐ’（ｘ）＝α_1・ｘｍｏｄｑ）を、サーバの数がｎ個だったらｎ次多項式（ｐ’（ｘ）＝α_1・ｘ＋α_2・x^2＋・・・＋α_n・x^nｍｏｄｑ）を発生する。ここで、αは（Ｚ／ｑＺ）*から無作為に選ばれる。例えば、一つのサーバの場合、ｐ’（ｘ）は、ｐ’（ｘ）＝α_1・ｘｍｏｄｑとなる。ここで、ユーザは自分が覚えているパスワード（例えば、”Ｐｏｏｈ９３”）を入力する。多項式とユーザのパスワードが入力されたらパスワード認証データ生成器１１２は、パスワード認証データＨを生成する。パスワード認証データＨは、例えば$H = h^{p'(1)+Pooh^{93}} \mod p$により計算できる。ここで、ｐ’（１）はｐ’（ｘ）でｘの代わりにサーバのＩＤ（例えば、「１」）を入れて計算した値である。パスワード認証データＨはユーザが直接にサーバに渡したり、郵便で送付したり、あるいは電話で知らせるなどして、安全に通知する必要がある。ユーザの端末装置の内部にあるメモリ１２は多項式発生器から発生された多項式Ｐ’＝ｐ’（ｘ）を記憶して保存する。

（２）多項式を利用する場合（その２）

次に、図２を参照して、多項式を利用する場合（その２）について説明する。

まず、多項式発生器１１１によりランダムに多項式を発生する。このとき、

登録するサーバの数が一つだったらｘを変数とする１次多項式（ｐ’（ｘ）＝α_1・ｘｍｏｄｑ）を、サーバの数がｎ個だったらｎ次多項式（ｐ’（ｘ）＝α_1・ｘ＋α_2・x^2＋・・・＋α_n・x^nｍｏｄｑ）を発生する。ここで、αは（Ｚ／ｑＺ）*から無作為に選ばれる。例えば、一つのサーバの場合、ｐ’（ｘ）は、ｐ’（ｘ）＝α_1・ｘｍｏｄｑとなる。ここで、ユーザは自分が覚えているパスワード（例えば、”Ｐｏｏｈ９３”）を入力する。多項式とユーザのパスワードが入力されたらパスワード認証データ生成器１１２は、パスワード認証データＨを生成する。パスワード認証データＨは、例えばＨ＝ｐ（１）＝ｐ’（１）＋Ｐｏｏｈ９３ｍｏｄｑにより計算できる。ここで、ｐ’（１）はｐ’（ｘ）でｘの代わりにサーバのＩＤ（例えば、「１」）を入れて計算した値である。パスワード認証データＨはユーザが直接にサーバに渡したり、郵便で送付したり、あるいは電話で知らせるなどして、安全に通知する必要がある。ユーザの端末装置の内部にあるメモリ１２は多項式発生器から発生された多項式Ｐ’＝ｐ’（ｘ）を記憶して保存する。

（３）多項式とハッシュ関数を利用する場合（その１）

次に、図８を参照して、多項式とハッシュ関数を利用する場合（その１）について説明する。

まず、多項式発生器１１９によりランダムに多項式を発生する。このとき、登録するサーバの数が一つだったらｘを変数とする１次多項式（ｐ’（ｘ）＝α_1・ｘｍｏｄＮ）を、サーバの数がｎ個だったらｎ次多項式（ｐ’（ｘ）＝α_1・ｘ＋α_2・x^2＋・・・＋α_n・x^nｍｏｄＮ）を発生する。ここで、αは（Ｚ／ｑＺ）*から無作為に選ばれる。例えば、一つのサーバの場合、ｐ’（ｘ）は、ｐ’（ｘ）＝α_1・ｘｍｏｄＮとなる。そして、ハッシュ関数発生器１２０によりランダムにハッシュ関数ＨＡＳＨを発生する。ＨＡＳＨは一方向ハッシュ関数である。ここで、ユーザは自分が覚えているパスワード（例えば、”Ｐｏｏｈ９３”）を入力する。多項式とハッシュ関数とユーザのパスワードが入力されたらパスワード認証データ生成器１２１は、パスワード認証データＨを生成する。パスワード認証データＨは、例えばＨ＝$h^{p(1)}$ｍｏｄｐにより計

算できる。ここでｐ（１）は、ｐ（１）＝ｐ’（１）＋ＨＡＳＨ（Ｐｏｏｈ９３‖ＩＤ（Ｕ）‖ＩＤ（Ｓ））ｍｏｄＮにより計算する。ここで、ＩＤ（Ｕ）とＩＤ（Ｓ）はそれぞれユーザとサーバのＩＤを表す。ここで、ｐ’（１）はｐ’（ｘ）でｘの代わりに「１」入れて計算した値である。

例えば、登録するサーバの数がｎの場合、パスワード認証データ生成器１２１は、ｉ番目のサーバに対してパスワード認証データＨを生成する。パスワード認証データＨは、例えば$H = h^{p(1)} \bmod p$により計算できる。ここでｐ（ｉ）は、ｐ（ｉ）＝ｐ’（ｉ）＋ＨＡＳＨ（Ｐｏｏｈ９３‖ＩＤ（Ｕ）‖ＩＤ（Ｓ））ｍｏｄＮにより計算する。ここで、ＩＤ（Ｕ）とＩＤ（Ｓ）はそれぞれユーザとｉ番目のサーバのＩＤを表す。ここで、ｐ’（ｉ）はｎ次多項式ｐ’（ｘ）でｘの代わりに「ｉ」を入れて計算した値である。

パスワード認証データＨはユーザが直接にサーバに渡したり、郵便で送付したり、あるいは電話で知らせるなどして、安全に通知する必要がある。ユーザの端末装置の内部にあるメモリ１２は多項式発生器から発生された多項式ｐ’（ｘ）とハッシュ関数発生器から発生されたハッシュ関数ＨＡＳＨとを一緒にＰ’＝（ｐ’（ｘ），ＨＡＳＨ）として記憶して保存する。

（４）多項式とハッシュ関数を利用する場合（その２）

次に、図８を参照して、多項式とハッシュ関数を利用する場合（その2）について説明する。

まず、多項式発生器１１９によりランダムに多項式を発生する。このとき、登録するサーバの数が一つだったらｘを変数とする１次多項式（$p'(x) = \alpha_1 \cdot x \bmod N$）を、サーバの数がｎ個だったらｎ次多項式（$p'(x) = \alpha_1 \cdot x + \alpha_2 \cdot x^2 + \cdots + \alpha_n \cdot x^n \bmod N$）を発生する。ここで、$\alpha$は$(Z/qZ)^*$から無作為に選ばれる。例えば、一つのサーバの場合、ｐ’（ｘ）は、ｐ’（ｘ）＝$\alpha_1 \cdot x \bmod N$となる。そして、ハッシュ関数発生器１２０によりランダムにハッシュ関数ＨＡＳＨを発生する。ＨＡＳＨは一方向ハッシュ関数である。ここで、ユーザは自分が覚えているパスワード（例えば、”Ｐｏｏｈ９３”）を入力する。多項式とハッシュ関数とユーザのパスワードが

入力されたらパスワード認証データ生成器１２１は、パスワード認証データＨを生成する。パスワード認証データＨは、例えばＨ＝ｐ（１）＝ｐ’（１）＋ＨＡＳＨ（Ｐｏｏｈ９３‖ＩＤ（Ｕ）‖ＩＤ（Ｓ））ｍｏｄＮにより計算できる。ここで、ＩＤ（Ｕ）とＩＤ（Ｓ）はそれぞれユーザとサーバのＩＤを表す。ここで、ｐ’（１）はｐ’（ｘ）でｘの代わりに「１」入れて計算した値である。

例えば、登録するサーバの数がｎの場合、パスワード認証データ生成器１２１は、ｉ番目のサーバに対してパスワード認証データＨを生成する。パスワード認証データＨは、例えばＨ＝ｐ（ｉ）＝ｐ’（ｉ）＋ＨＡＳＨ（Ｐｏｏｈ９３‖ＩＤ（Ｕ）‖ＩＤ（Ｓ））ｍｏｄＮにより計算できる。ここで、ＩＤ（Ｕ）とＩＤ（Ｓ）はそれぞれユーザとｉ番目のサーバのＩＤを表す。ここで、ｐ’（ｉ）はｎ次多項式ｐ’（ｘ）でｘの代わりに「ｉ」を入れて計算した値である。

パスワード認証データＨはユーザが直接にサーバに渡したり、郵便で送付したり、あるいは電話で知らせるなどして、安全に通知する必要がある。ユーザの端末装置の内部にあるメモリ１２は多項式発生器から発生された多項式ｐ’（ｘ）とハッシュ関数発生器から発生されたハッシュ関数ＨＡＳＨとを一緒にＰ’＝（ｐ’（ｘ），ＨＡＳＨ）として記憶して保存する。

（５）ハッシュ関数を利用する場合（その１）

次に、図３を参照して、ハッシュ関数を利用する場合（その１）について説明する。

まず、ハッシュ関数発生器１１３によりランダムにハッシュ関数ＨＡＳＨを発生する。そして、秘密値発生器１１４もランダムに秘密値Ｓを発生する。ここで、ユーザは自分が覚えているパスワード（例えば、”Ｐｏｏｈ９３”）を入力する。ハッシュ関数ＨＡＳＨと秘密値Ｓとユーザのパスワードが入力されたらパスワード認証データ生成器１１５はパスワード認証データＨを生成する。パスワード認証データＨは、例えば、$H = h^{HASH(S \| Pooh93 \| ID(U) \| ID(S))}$ ｍｏｄｐにより計算できる。ここで、ＩＤ（Ｕ）とＩＤ（Ｓ）はそれぞれユー

ザとサーバのＩＤを表す。パスワード認証データＨはユーザが直接にサーバに渡したり、郵便で送付したり、あるいは電話で知らせるなどして、安全に通知する必要がある。ユーザの端末装置の内部にあるメモリ１２はハッシュ関数発生器１１３と秘密値発生器１１４から発生されたハッシュ関数ＨＡＳＨと秘密値Ｓとを一緒にＰ’＝（Ｓ，ＨＡＳＨ）として記憶して保存する。

（６）ハッシュ関数を利用する場合（その２）

次に、図３を参照して、ハッシュ関数を利用する場合（その２）について説明する。

まず、ハッシュ関数発生器１１３によりランダムにハッシュ関数ＨＡＳＨを発生する。そして、秘密値発生器１１４もランダムに秘密値Ｓを発生する。ここで、ユーザは自分が覚えているパスワード（例えば、”Ｐｏｏｈ９３”）を入力する。ハッシュ関数ＨＡＳＨと秘密値Ｓとユーザのパスワードが入力されたらパスワード認証データ生成器１１５はパスワード認証データＨを生成する。パスワード認証データＨは、例えば、Ｈ＝ＨＡＳＨ（Ｓ‖Ｐｏｏｈ９３‖ＩＤ（Ｕ）‖ＩＤ（Ｓ））ｍｏｄｑにより計算できる。ここで、ＩＤ（Ｕ）とＩＤ（Ｓ）はそれぞれユーザとサーバのＩＤを表す。パスワード認証データＨはユーザが直接にサーバに渡したり、郵便で送付したり、あるいは電話で知らせるなどして、安全に通知する必要がある。ユーザの端末装置の内部にあるメモリ１２はハッシュ関数発生器１１３と秘密値発生器１１４から発生されたハッシュ関数ＨＡＳＨと秘密値Ｓとを一緒にＰ’＝（Ｓ，ＨＡＳＨ）として記憶して保存する。

（７）擬似乱数発生器を利用する場合（その１）

次に、図４を参照して、擬似乱数発生器を利用する場合（その１）について説明する。

まず、擬似乱数発生器１１６によりランダムに擬似乱数関数ＰＲＮＧを発生する。そして、秘密値発生器１１７もランダムに秘密値Ｓを発生する。ここで、ユーザは自分が覚えているパスワード（例えば、”Ｐｏｏｈ９３”）を入力す

る。擬似乱数関数ＰＲＮＧと秘密値Ｓとユーザのパスワードが入力されたのを受けて、パスワード認証データ生成器１１８はパスワード認証データＨを生成する。パスワード認証データＨは、例えば、$H = h^{PRNG(S \| Pooh93 \| ID(U) \| ID(S))} \bmod p$により計算できる。ここで、ＩＤ（Ｕ）とＩＤ（Ｓ）はそれぞれユーザとサーバのＩＤを表す。パスワード認証データＨはユーザが直接にサーバに渡したり、郵便で送付したり、あるいは電話で知らせるなどして、安全に通知する必要がある。ユーザの端末装置の内部にあるメモリ１２は擬似乱数発生器１１６と秘密値発生器１１７から発生させた擬似乱数関数ＰＲＮＧと秘密値Ｓとを一緒にＰ’＝（Ｓ，ＰＲＮＧ）として記憶して保存する。

（８）擬似乱数発生器を利用する場合（その２）

次に、図４を参照して、擬似乱数発生器を利用する場合（その２）について説明する。

まず、擬似乱数発生器１１６によりランダムに擬似乱数関数ＰＲＮＧを発生する。そして、秘密値発生器１１７もランダムに秘密値Ｓを発生する。ここで、ユーザは自分が覚えているパスワード（例えば、”Ｐｏｏｈ９３”）を入力する。擬似乱数関数ＰＲＮＧと秘密値Ｓとユーザのパスワードが入力されたのを受けて、パスワード認証データ生成器１１８はパスワード認証データＨを生成する。パスワード認証データＨは、例えば、Ｈ＝ＰＲＮＧ（Ｓ‖Ｐｏｏｈ９３‖ＩＤ（Ｕ）‖ＩＤ（Ｓ））ｍｏｄｑにより計算できる。ここで、ＩＤ（Ｕ）とＩＤ（Ｓ）はそれぞれユーザとサーバのＩＤを表す。パスワード認証データＨはユーザが直接にサーバに渡したり、郵便で送付したり、あるいは電話で知らせるなどして、安全に通知する必要がある。ユーザの端末装置の内部にあるメモリ１２は擬似乱数発生器１１６と秘密値発生器１１７から発生させた擬似乱数関数ＰＲＮＧと秘密値Ｓとを一緒にＰ’＝（Ｓ，ＰＲＮＧ）として記憶して保存する。

次に、図６、７を参照して、前述した初期化を行った端末装置１とサーバ２（図５参照）との間で相互認証及び鍵交換を行う動作を説明する。

＜端末装置の動作＞

（１）多項式を利用した場合（その１とその２）

初めに、多項式を利用した場合の端末装置１の動作を説明する。端末装置１は、前述した多項式を利用した場合（その１）、多項式を利用した場合（その２）に関わらず次のように動作する。

まず、ユーザの端末装置１に備えたメモリ１２から記憶された多項式Ｐ’＝ｐ’（ｘ）を読み出す。結合器３２はメモリ１２から読み出した多項式Ｐ’とユーザが入力したパスワードによりＰ＝ｐ（ｘ）を計算して出力する。例えば、ｐ（ｘ）＝ｐ’（ｘ）＋Ｐｏｏｈ９３＝α_1・ｘ＋Ｐｏｏｈ９３ｍｏｄｑにより計算する。マスク演算器３４は、結合器３２から入力されたＰと乱数発生器３３において発生させた乱数R_1とからY_1を、$Y_1=g^{R1}\cdot h^{-p(1)}$ｍｏｄｐにより計算する。ここでｐ（１）は、ｐ（１）＝ｐ’（１）＋Ｐｏｏｈ９３＝α_1・１＋Ｐｏｏｈ９３ｍｏｄｑにより計算する。ここで、「１」はサーバの認証ＩＤを表す。通信処理部３５はY_1をサーバ２へ送信し、サーバ２からY_2を受信する。マスター鍵生成器３６は結合器３２から出力されるＰと乱数発生器３３から出力されるR_1と受信したY_2を入力としてＭＫを、$MK=(Y_2\cdot h^{-p(1)})^{R1}$ ｍｏｄｐにより計算して出力する。

続いて認証結果判断部３７は、ＭＫを入力として、V_1＝ＨＡＳＨ（００‖Y_1‖Y_2‖ＭＫ）によりV_1を計算してこのV_1を通信処理部３５によりサーバ２へ送信し、サーバ２から受信したV_2とＨＡＳＨ（０１‖Y_1‖Y_2‖ＭＫ）を比較する。ここで、ＨＡＳＨは一方向ハッシュ関数であるし、ＨＡＳＨの代わりにＭＡＣ（Ｍｅｓｓａｇｅ Ａｕｔｈｅｎｔｉｃａｔｉｏｎ Ｃｏｄｅ）を使ってもよい。

次に、認証結果判断部３７においてV_2とＨＡＳＨ（０１‖Y_1‖Y_2‖ＭＫ）が一致しない場合、認証結果判断部３７は、エラー発生器３８に対して、一致しないことを通知する。これを受けて、エラー発生器３８はエラーを発生して処理を中断する。一方、認証結果判断部３７においてV_2とＨＡＳＨ（０１‖Y_1‖Y_2‖ＭＫ）が一致した場合はサーバ２が正当な装置として認証してセッション鍵生成器３９は、ＳＫ＝ＨＡＳＨ（１１‖Y_1‖Y_2‖ＭＫ）によりセッ

ション鍵ＳＫを生成する。

（２）多項式とハッシュ関数を利用した場合（その１とその２）

次に、多項式とハッシュ関数を利用した場合の端末装置１の動作を説明する。端末装置１は、前述した多項式とハッシュ関数を利用した場合（その１）、多項式とハッシュ関数を利用した場合（その２）に関わらず次のように動作する。

まず、ユーザの端末装置１に備えたメモリ１２から記憶された多項式とハッシュ関数Ｐ’＝（ｐ’（ｘ），ＨＡＳＨ）を読み出す。結合器３２はメモリ１２から読み出した多項式ｐ’（ｘ）とハッシュ関数ＨＡＳＨとユーザが入力したパスワードによりＰ＝ｐ（ｘ）を計算して出力する。例えば、ｐ’（ｘ）が１次多項式の場合、ｐ（ｘ）＝ｐ’（ｘ）＋ＨＡＳＨ（Ｐｏｏｈ９３‖ＩＤ（Ｕ）‖ＩＤ（Ｓ））＝α_1・ｘ＋ＨＡＳＨ（Ｐｏｏｈ９３‖ＩＤ（Ｕ）‖ＩＤ（Ｓ））ｍｏｄＮにより計算する。マスク演算器３４は、結合器３２から入力されたＰと乱数発生器３３において発生させた乱数R_1とからY_1を、$Y_1=g^{R1}\cdot h^{-p(1)}$ ｍｏｄｐにより計算する。ここでｐ（１）は、ｐ（１）＝ｐ’（１）＋ＨＡＳＨ（Ｐｏｏｈ９３‖ＩＤ（Ｕ）‖ＩＤ（Ｓ））＝α_1・１＋ＨＡＳＨ（Ｐｏｏｈ９３‖ＩＤ（Ｕ）‖ＩＤ（Ｓ））ｍｏｄＮにより計算する。ここで、ｐ’（１）はｐ’（ｘ）でｘの代わりに「１」入れて計算した値である。通信処理部３５はY_1をサーバ２へ送信し、サーバ２からY_2を受信する。マスター鍵生成器３６は結合器３２から出力されるＰと乱数発生器３３から出力されるR_1と受信したY_2を入力としてＭＫを、$MK=(Y_2\cdot h^{-p(1)})^{R1}$ ｍｏｄｐにより計算して出力する。

ユーザの端末装置１に備えたメモリ１２から読み出した多項式ｐ’（ｘ）がｎ次多項式の場合、結合器３２は多項式ｐ’（ｘ）とハッシュ関数ＨＡＳＨとユーザが入力したパスワードによりＰ＝ｐ（ｘ）を計算して出力する。例えば、ｐ（ｘ）＝ｐ’（ｘ）＋ＨＡＳＨ（Ｐｏｏｈ９３‖ＩＤ（Ｕ）‖ＩＤ（Ｓ））ｍｏｄＮにより計算する。マスク演算器３４は、結合器３２から入力されたＰと乱数発生器３３において発生させた乱数R_1とからY_1を、$Y_1=g^{R1}\cdot h^{-p(1)}$ ｍｏｄｐにより計算する。ここでｐ（ｉ）は、ｐ（ｉ）＝ｐ’（ｉ）＋ＨＡＳ

H（Pooh93‖ID（U）‖ID（S））modNにより計算する。ここで、p'（i）はi番目のサーバに対してp'（x）でxの代わりに「i」入れて計算した値である。通信処理部35はY_1をサーバ2へ送信し、サーバ2からY_2を受信する。マスター鍵生成器36は結合器32から出力されるPと乱数発生器33から出力されるR_1と受信したY_2を入力としてMKを、MK＝$(Y_2 \cdot h^{-p(i)})^{R1}$ modpにより計算して出力する。

続いて認証結果判断部37は、MKを入力として、V_1＝HASH（00‖Y_1‖Y_2‖MK）によりV_1を計算してこのV_1を通信処理部35によりサーバ2へ送信し、サーバ2から受信したV_2とHASH（01‖Y_1‖Y_2‖MK）を比較する。ここで、HASHは一方向ハッシュ関数であるし、HASHの代わりにMAC（Message Authentication Code）を使ってもよい。

次に、認証結果判断部37においてV_2とHASH（01‖Y_1‖Y_2‖MK）が一致しない場合、認証結果判断部37は、エラー発生器38に対して、一致しないことを通知する。これを受けて、エラー発生器38はエラーを発生して処理を中断する。一方、認証結果判断部37においてV_2とHASH（01‖Y_1‖Y_2‖MK）が一致した場合はサーバ2が正当な装置として認証してセッション鍵生成器39は、SK＝HASH（11‖Y_1‖Y_2‖MK）によりセッション鍵SKを生成する。

（3）ハッシュ関数を利用した場合（その1とその2）

次に、ハッシュ関数を利用した場合の端末装置1の動作を説明する。端末装置1は、前述したハッシュ関数を利用した場合（その1）、ハッシュ関数を利用した場合（その2）に関わらず次のように動作する。

まず、ユーザの端末装置1に備えたメモリ12から記憶された秘密値とハッシュ関数P'＝（S，HASH）を読み出す。結合器32はメモリ12から読み出した秘密値Sとハッシュ関数HASHとユーザが入力したパスワードによりP＝pを計算して出力する。例えば、pは、p＝HASH（S‖Pooh93‖ID（U）‖ID（S））modqにより計算する。マスク演算器34

は、結合器３２から入力されたＰと乱数発生器３３において発生させた乱数Ｒ$_1$とからＹ$_1$を、$Y_1 = g^{R1} \cdot h^{-p} \bmod p$により計算する。通信処理部３５は、Ｙ$_1$をサーバ２へ送信し、サーバ２からＹ$_2$を受信する。マスター鍵生成器３６は結合器３２から出力されるＰと乱数発生器３３から出力されるＲ$_1$と受信したＹ$_2$を入力としてＭＫを、$MK = (Y_2 \cdot h^{-p})^{R1} \bmod p$により計算して出力する。

続いて認証結果判断部３７は、ＭＫを入力として、Ｖ$_1$＝ＨＡＳＨ（００‖Ｙ$_1$‖Ｙ$_2$‖ＭＫ）によりＶ$_1$を計算してこのＶ$_1$を通信処理部３５によりサーバ２へ送信し、サーバ２から受信したＶ$_2$とＨＡＳＨ（０１‖Ｙ$_1$‖Ｙ$_2$‖ＭＫ）を比較する。ここで、ＨＡＳＨは一方向ハッシュ関数であるし、ＨＡＳＨの代わりにＭＡＣ（Ｍｅｓｓａｇｅ　Ａｕｔｈｅｎｔｉｃａｔｉｏｎ　Ｃｏｄｅ）を使ってもよい。

次に、認証結果判断部３７においてＶ$_2$とＨＡＳＨ（０１‖Ｙ$_1$‖Ｙ$_2$‖ＭＫ）が一致しない場合、認証結果判断部３７は、エラー発生器３８に対して、一致しないことを通知する。これを受けて、エラー発生器３８はエラーを発生して処理を中断する。一方、認証結果判断部３７においてＶ$_2$とＨＡＳＨ（０１‖Ｙ$_1$‖Ｙ$_2$‖ＭＫ）が一致した場合はサーバ２が正当な装置として認証してセッション鍵生成器３９は、ＳＫ＝ＨＡＳＨ（１１‖Ｙ$_1$‖Ｙ$_2$‖ＭＫ）によりセッション鍵ＳＫを生成する。

（４）擬似乱数発生器を利用した場合（その１とその２）

次に、疑似乱数関数を利用した場合の端末装置１の動作を説明する。端末装置１は、前述した擬似乱数発生器を利用した場合（その１）、擬似乱数発生器を利用した場合（その２）に関わらず次のように動作する。

疑似乱数関数を利用した場合は、ユーザの端末装置１に備えたメモリ１２に記憶されたハッシュ関数ＨＡＳＨの代わりに疑似乱数関数ＰＲＮＧを用いる以外は、ハッシュ関数を利用した場合と同様の動作であるため、ここでは詳細な説明を省略する。

＜サーバの動作＞

（１）多項式を利用した場合（その１）、多項式とハッシュ関数を利用した場合（その１）、ハッシュ関数を利用した場合（その１）、擬似乱数関数を利用した場合（その１）

サーバ２は、前述した多項式利用した場合（その１）、多項式とハッシュ関数を利用した場合（その１）、ハッシュ関数利用した場合（その１）、擬似乱数発生器を利用した場合（その１）に関わらず次のように動作する。

サーバ２に備えたメモリ４１に保存されたユーザＩＤとパスワードの認証データＨを読み出す。マスク演算器４３はメモリ４１から読み出したＨと乱数発生器４２から発生させた乱数R_2を入力としてY_2を、$Y_2 = g^{R2} \cdot H \bmod p$により計算する。通信処理部４４は、計算して得られたY_2を端末装置１に送信し、端末装置１から受信したY_1をマスター鍵生成器４５へ出力する。マスク鍵生成器４５はメモリ４１から読み出したＨと乱数発生器４２からのR_2と通信処理部４４からのY_1を入力としてＭＫを、$MK = (Y_1 \cdot H)^{R2} \bmod p$により計算して、ＭＫを出力する。

続いて認証結果判断部４６は、ＭＫを入力として、V_2＝ＨＡＳＨ（０１‖Y_1‖Y_2‖ＭＫ）によりV_2を計算してこのV_2を通信処理部４４により端末装置１へ送信し、端末装置１から受信したV_1とＨＡＳＨ（００‖Y_1‖Y_2‖ＭＫ）を比較する。ここで、ＨＡＳＨは一方向ハッシュ関数であるし、ＨＡＳＨの代わりにＭＡＣ（Ｍｅｓｓａｇｅ　Ａｕｔｈｅｎｔｉｃａｔｉｏｎ　Ｃｏｄｅ）を使ってもよい。

次に、認証結果判断部４６においてV_1とＨＡＳＨ（００‖Y_1‖Y_2‖ＭＫ）が一致しない場合、認証結果判断部４６は、エラー発生器４７に対して、一致しないことを通知する。これを受けて、エラー発生器４７はエラーを発生して処理を中断する。一方、認証結果判断部４６においてV_1とＨＡＳＨ（００‖Y_1‖Y_2‖ＭＫ）が一致した場合は端末装置１が正当な装置として認証してセッション鍵生成器４８は、ＳＫ＝ＨＡＳＨ（１１‖Y_1‖Y_2‖ＭＫ）によりセッション鍵ＳＫを生成する。

（２）多項式を利用した場合（その２）、多項式とハッシュ関数を利用した場合（その２）、ハッシュ関数を利用した場合（その２）、擬似乱数関数を利用した場合（その２）

サーバ２は、前述した多項式利用した場合（その２）、多項式とハッシュ関数を利用した場合（その２）、ハッシュ関数を利用した場合（その２）、擬似乱数発生器を利用した場合（その２）に関わらず次のように動作する。

サーバ２に備えたメモリ４１に保存されたユーザＩＤとパスワードの認証データＨを読み出す。マスク演算器４３はメモリ４１から読み出したＨと乱数発生器４２から発生させた乱数R_2を入力としてY_2を、$Y_2=g^{R2}\cdot h^{H} \bmod p$により計算する。通信処理部４４は、計算して得られたY_2を端末装置１に送信し、端末装置１から受信したY_1をマスター鍵生成器４５へ出力する。マスク鍵生成器４５はメモリ４１から読み出したＨと乱数発生器４２からのR_2と通信処理部４４からのY_1を入力としてＭＫを、$MK=(Y_1\cdot h^{H})^{R2} \bmod p$により計算して、ＭＫを出力する。

続いて認証結果判断部４６は、ＭＫを入力として、V_2＝ＨＡＳＨ（０１‖Y_1‖Y_2‖ＭＫ）によりV_2を計算してこのV_2を通信処理部４４により端末装置１へ送信し、端末装置１から受信したV_1とＨＡＳＨ（００‖Y_1‖Y_2‖ＭＫ）を比較する。ここで、ＨＡＳＨは一方向ハッシュ関数であるし、ＨＡＳＨの代わりにＭＡＣ（Ｍｅｓｓａｇｅ　Ａｕｔｈｅｎｔｉｃａｔｉｏｎ　Ｃｏｄｅ）を使ってもよい。

次に、認証結果判断部４６においてV_1とＨＡＳＨ（００‖Y_1‖Y_2‖ＭＫ）が一致しない場合、認証結果判断部４６は、エラー発生器４７に対して、一致しないことを通知する。これを受けて、エラー発生器４７はエラーを発生して処理を中断する。一方、認証結果判断部４６においてV_1とＨＡＳＨ（００‖Y_1‖Y_2‖ＭＫ）が一致した場合は端末装置１が正当な装置として認証してセッション鍵生成器４８は、ＳＫ＝ＨＡＳＨ（１１‖Y_1‖Y_2‖ＭＫ）によりセッション鍵ＳＫを生成する。

＜パスワード認証データの更新－１＞

ユーザは、サーバに対してすでに登録されたパスワード認証データを、自分が覚えているパスワードを変えずに、更新したい時、自分の端末装置の更新化を行う。図９は、ユーザの端末装置の更新化処理の構成を示すブロック図である。更新化処理は、多項式発生器１３による多項式Ｔ’とユーザの端末装置１に備えたメモリ１２から記憶された多項式Ｐ’が入力されると、更新値生成器１４によって、サーバ更新用の値Ｈ’と、ユーザ保存用の更新された多項式Ｐ’が生成され、Ｈ’は、サーバに受け渡され、更新された多項式Ｐ’は、メモリ１２へ保存する。ここでの更新化処理は、前述した多項式を利用した場合（その１）、多項式を利用した場合（その２）、多項式とハッシュ関数を利用した場合（その１）、多項式とハッシュ関数を利用した場合（その２）に適用することが可能である。

＜端末装置の更新化処理＞

（１）多項式を利用した場合（その１）

初めに、図９を参照して、多項式を利用した場合（その１）の端末装置１の更新化処理の動作を説明する。

まず、多項式発生器１３によりランダムに多項式を発生する。このとき、登録したサーバの数が一つだったらｘを変数とする１次多項式（ｔ’（ｘ）＝β_1・ｘｍｏｄｑ）を、サーバの数がｎ個だったらｎ次多項式（ｔ’（ｘ）＝β_1・ｘ＋β_2・x^2＋・・・＋β_n・x^nｍｏｄｑ）を発生する。ここで、βは（Ｚ／ｑＺ）*から無作為に選ばれる。例えば、一つのサーバの場合、Ｔ’＝ｔ’（ｘ）は、ｔ’（ｘ）＝β_1・ｘｍｏｄｑとなる。ここで、ユーザの端末装置１に備えたメモリ１２から記憶された多項式Ｐ’＝ｐ’（ｘ）を読み出す。多項式ｔ’（ｘ）と多項式ｐ’（ｘ）が入力されたら更新値生成器１４は、ユーザ保存用の更新された多項式Ｐ’とサーバ更新用の値Ｈ’を生成する。更新された多項式Ｐ’は、例えばＰ’＝ｔ’（ｘ）＋ｐ’（ｘ）＝（α_1＋β_1）・ｘｍｏｄｑにより計算できる。サーバ更新用の値Ｈ’、例えばＨ’＝$h^{t'(1)}$ｍｏｄｐにより計算できる。ここで、ｔ’（１）はｔ’（ｘ）でｘの代わりにサーバのＩＤ（例えば、「１」）入れて計算した値である。サーバ更新用の値Ｈ’はユ

ーザが直接にサーバに渡したり、郵便で送付したり、あるいは電話で知らせるなどして、安全に通知する必要がある。ユーザの端末装置の内部にあるメモリ１２は更新された多項式Ｐ’ ＝ ｔ’（ｘ）＋ｐ’（ｘ）を記憶して保存する。

（２）多項式を利用した場合（その２）

次に、図９を参照して、多項式を利用した場合（その１）の端末装置１の更新化処理の動作を説明する。

まず、多項式発生器１３によりランダムに多項式を発生する。このとき、登録したサーバの数が一つだったらｘを変数とする１次多項式（ｔ’（ｘ）＝β_1・ｘｍｏｄｑ）を、サーバの数がｎ個だったらｎ次多項式（ｔ’（ｘ）＝β_1・ｘ＋β_2・x^2＋・・・＋β_n・x^nｍｏｄｑ）を発生する。ここで、βは（Ｚ／ｑＺ）*から無作為に選ばれる。例えば、一つのサーバの場合、Ｔ’＝ｔ’（ｘ）は、ｔ’（ｘ）＝β_1・ｘｍｏｄｑとなる。ここで、ユーザの端末装置１に備えたメモリ１２から記憶された多項式Ｐ’＝ｐ’（ｘ）を読み出す。多項式ｔ’（ｘ）と多項式ｐ’（ｘ）が入力されたら更新値生成器１４は、ユーザ保存用の更新された多項式Ｐ’とサーバ更新用の値Ｈ’を生成する。更新された多項式Ｐ’は、例えばＰ’＝ ｔ’（ｘ）＋ｐ’（ｘ）＝（α_1＋β_1）・ｘｍｏｄｑにより計算できる。サーバ更新用の値Ｈ’、例えばＨ’＝ ｔ’（１）ｍｏｄｑにより計算できる。ここで、ｔ’（１）はｔ’（ｘ）でｘの代わりにサーバのＩＤ（例えば、「１」）入れて計算した値である。サーバ更新用の値Ｈ’はユーザが直接にサーバに渡したり、郵便で送付したり、あるいは電話で知らせるなどして、安全に通知する必要がある。ユーザの端末装置の内部にあるメモリ１２は更新された多項式Ｐ’ ＝ ｔ’（ｘ）＋ｐ’（ｘ）を記憶して保存する。

（３）多項式とハッシュ関数を利用した場合（その１）

次に、図９を参照して、多項式とハッシュ関数を利用した場合（その１）の端末装置１の更新化処理の動作を説明する。

まず、多項式発生器１３によりランダムに多項式を発生する。このとき、登

録したサーバの数が一つだったらｘを変数とする１次多項式（ｔ’（ｘ）＝β_1・ｘｍｏｄＮ）を、サーバの数がｎ個だったらｎ次多項式（ｔ’（ｘ）＝β_1・ｘ＋β_2・x^2＋・・・＋β_n・x^nｍｏｄＮ）を発生する。ここで、βは（Ｚ／ｑＺ）*から無作為に選ばれる。例えば、一つのサーバの場合、Ｔ’＝ｔ’（ｘ）は、ｔ’（ｘ）＝β_1・ｘｍｏｄＮとなる。ここで、ユーザの端末装置１に備えたメモリ１２から記憶された多項式とハッシュ関数Ｐ’＝（ｐ’（ｘ），ＨＡＳＨ）を読み出す。多項式ｔ’（ｘ）と多項式ｐ’（ｘ）が入力されたら更新値生成器１４は、ユーザ保存用の更新された多項式Ｐ’とサーバ更新用の値Ｈ’を生成する。更新された多項式Ｐ’は、例えばＰ’＝ｔ’（ｘ）＋ｐ’（ｘ）＝（α_1＋β_1）・ｘｍｏｄＮにより計算できる。サーバ更新用の値Ｈ’、例えばＨ’＝$h^{t'(1)}$ｍｏｄｐにより計算できる。ここで、ｔ’（１）はｔ’（ｘ）でｘの代わりに「１」を入れて計算した値である。

例えば、登録したサーバの数がｎの場合、更新値生成器１４は、ｉ番目のサーバに対してサーバ更新用の値Ｈ’を生成する。サーバ更新用の値Ｈ’は、例えばＨ’＝$h^{t'(1)}$ｍｏｄｐにより計算できる。ここで、ｔ’（ｉ）はｎ次多項式ｔ’（ｘ）でｘの代わりに「ｉ」を入れて計算した値である。

サーバ更新用の値Ｈ’はユーザが直接にサーバに渡したり、郵便で送付したり、あるいは電話で知らせるなどして、安全に通知する必要がある。ユーザの端末装置の内部にあるメモリ１２は更新された多項式Ｐ’＝ｔ’（ｘ）＋ｐ’（ｘ）とメモリ１２から読み出したハッシュ関数ＨＡＳＨとを一緒にＰ’＝（ｔ’（ｘ）＋ｐ’（ｘ），ＨＡＳＨ）として記憶して保存する。

（４）多項式とハッシュ関数を利用した場合（その２）

次に、図９を参照して、多項式とハッシュ関数を利用した場合（その２）の端末装置１の更新化処理の動作を説明する。

まず、多項式発生器１３によりランダムに多項式を発生する。このとき、登録したサーバの数が一つだったらｘを変数とする１次多項式（ｔ’（ｘ）＝β_1・ｘｍｏｄＮ）を、サーバの数がｎ個だったらｎ次多項式（ｔ’（ｘ）＝β_1・ｘ＋β_2・x^2＋・・・＋β_n・x^nｍｏｄＮ）を発生する。ここで、βは（Ｚ／

ｑＺ）*から無作為に選ばれる。例えば、一つのサーバの場合、Ｔ’＝ｔ’（ｘ）は、ｔ’（ｘ）＝β_1・ｘｍｏｄＮとなる。ここで、ユーザの端末装置１に備えたメモリ１２から記憶された多項式とハッシュ関数Ｐ’＝（ｐ’（ｘ），ＨＡＳＨ）を読み出す。多項式ｔ’（ｘ）と多項式ｐ’（ｘ）が入力されたら更新値生成器１４は、ユーザ保存用の更新された多項式Ｐ’とサーバ更新用の値Ｈ’を生成する。更新された多項式Ｐ’は、例えばＰ’＝ｔ’（ｘ）＋ｐ’（ｘ）＝（$\alpha_1+\beta_1$）・ｘｍｏｄＮにより計算できる。サーバ更新用の値Ｈ’、例えばＨ’＝ｔ’（１）ｍｏｄＮにより計算できる。ここで、ｔ’（１）はｔ’（ｘ）でｘの代わりに「１」を入れて計算した値である。

例えば、登録したサーバの数がｎの場合、更新値生成器１４は、ｉ番目のサーバに対してサーバ更新用の値Ｈ’を生成する。サーバ更新用の値Ｈ’は、例えばＨ’＝ｔ’（ｉ）ｍｏｄＮにより計算できる。ここで、ｔ’（ｉ）はｎ次多項式ｔ’（ｘ）でｘの代わりに「ｉ」を入れて計算した値である。

サーバ更新用の値Ｈ’はユーザが直接にサーバに渡したり、郵便で送付したり、あるいは電話で知らせるなどして、安全に通知する必要がある。ユーザの端末装置の内部にあるメモリ１２は更新された多項式Ｐ’＝ｔ’（ｘ）＋ｐ’（ｘ）とメモリ１２から読み出したハッシュ関数ＨＡＳＨとを一緒にＰ’＝（ｔ’（ｘ）＋ｐ’（ｘ），ＨＡＳＨ）として記憶して保存する。

＜サーバの更新化処理＞

（１）多項式を利用した場合（その１）、多項式とハッシュ関数を利用した場合（その１）

初めに、図１０を参照して、多項式を利用した場合（その１）、多項式とハッシュ関数を利用した場合（その１）のサーバ２の更新化処理の動作を説明する。サーバ２は、前述した多項式利用した場合（その１）、多項式とハッシュ関数を利用した場合（その１）に関わらず次のように動作する。

まず、サーバ２に備えたメモリ４１に保存されたユーザＩＤとパスワード認証データＨを読み出す。ユーザの端末装置から送られたサーバ更新用の値Ｈ’とメモリ４１から読み出したパスワード認証データＨが入力されたら更新値

生成器２１は、サーバ保存用の更新されたパスワード認証データＨを生成する。更新されたパスワード認証データＨは、例えば$H = h^{p(1)} \cdot h^{t'(1)} = h^{p(1)+t'(1)} \bmod p$により計算できる。サーバの内部にあるメモリ４１は更新されたパスワード認証データＨを記憶して保存する。

（２）多項式を利用した場合（その２）

次に、図１０を参照して、前述した多項式を利用した場合（その２）のサーバ２の更新化処理の動作を説明する。

まず、サーバ２に備えたメモリ４１に保存されたユーザＩＤとパスワード認証データＨを読み出す。ユーザの端末装置から送られたサーバ更新用の値Ｈ’とメモリ４１から読み出したパスワード認証データＨが入力されたら更新値生成器２１は、サーバ保存用の更新されたパスワード認証データＨを生成する。更新されたパスワード認証データＨは、例えばＨ＝ ｐ（１）＋ｔ’（１）ｍｏｄｑにより計算できる。サーバの内部にあるメモリ４１は更新されたパスワード認証データＨを記憶して保存する。

（３）多項式とハッシュ関数を利用した場合（その２）

次に、図１０を参照して、前述した多項式とハッシュ関数を利用した場合（その２）のサーバ２の更新化処理の動作を説明する。

まず、サーバ２に備えたメモリ４１に保存されたユーザＩＤとパスワード認証データＨを読み出す。ユーザの端末装置から送られたサーバ更新用の値Ｈ’とメモリ４１から読み出したパスワード認証データＨが入力されたら更新値生成器２１は、サーバ保存用の更新されたパスワード認証データＨを生成する。更新されたパスワード認証データＨは、例えばＨ＝ ｐ（１）＋ｔ’（１）ｍｏｄＮにより計算できる。サーバの内部にあるメモリ４１は更新されたパスワード認証データＨを記憶して保存する。

＜パスワード認証データの更新－２＞

ユーザは、サーバに対してすでに登録されたパスワード認証データを、自分

が覚えているパスワードを変えながら、更新したい時、自分の端末装置の更新化処理を行う。図１１は、ユーザの端末装置の更新化処理の構成を示すブロック図である。更新化処理は、秘密値発生器１５による秘密値Ｓ’とユーザの新しいパスワードＰＷ’とユーザの端末装置１に備えたメモリ１２から記憶されたＰ’が入力されると、パスワード認証データ更新器１６によって、サーバ更新用のパスワード認証データＨ’と、ユーザ保存用の更新されたＰ’が生成され、Ｈ’は、サーバ２に受け渡され、更新されたＰ’は、メモリ１２へ保存する。ここでの更新化処理は、前述したハッシュ関数を利用した場合（その１）、ハッシュ関数を利用した場合（その２）、擬似乱数発生器を利用した場合（その１）、擬似乱数発生器を利用した場合（その２）に適用することが可能である。そして、多項式を利用した場合（その１）、多項式を利用した場合（その２）、多項式とハッシュ関数を利用した場合（その１）、多項式とハッシュ関数を利用した場合（その２）も同じように初期化処理と同様の動作を用いて適用することが可能なため、ここでは詳細な説明を省略する。

＜端末装置の更新化処理＞

（１）ハッシュ関数を利用した場合（その１）

初めに、図１１を参照して、ハッシュ関数を利用した場合（その１）の端末装置１の更新化処理の動作を説明する。

まず、秘密値発生器１５によりランダムに秘密値Ｓ’を発生する。ここで、ユーザの端末装置１に備えたメモリ１２から記憶されたＰ’＝（Ｓ，ＨＡＳＨ）を読み出す。ユーザが覚えている新しいパスワード（ＰＷ’）とハッシュ関数ＨＡＳＨと秘密値Ｓ’が入力されたらパスワード認証データ更新器１６は、ユーザ保存用の更新されたＰ’とサーバ更新用のパスワード認証データＨ’を生成する。サーバ更新用のパスワード認証データＨ’は、例えば$H' = h^{HASH(S' \| PW' \| ID(U) \| ID(S))} \bmod p$により計算できる。ここで、ＩＤ（Ｕ）とＩＤ（Ｓ）はそれぞれユーザとサーバのＩＤを表す。サーバ更新用のパスワード認証データＨ’はユーザが直接にサーバ２に渡したり、郵便で送付したり、あるいは電話で知らせるなどして、安全に通知する必要がある。ユーザの端末装置１の内

部にあるメモリ１２は更新されたＰ’＝（Ｓ’，ＨＡＳＨ）を記憶して保存する。

（２）ハッシュ関数を利用した場合（その２）

次に、図１１を参照して、ハッシュ関数を利用した場合（その２）の端末装置１の更新化処理の動作を説明する。

まず、秘密値発生器１５によりランダムに秘密値Ｓ’を発生する。ここで、ユーザの端末装置１に備えたメモリ１２から記憶されたＰ’＝（Ｓ，ＨＡＳＨ）を読み出す。ユーザが覚えている新しいパスワード（ＰＷ’）とハッシュ関数ＨＡＳＨと秘密値Ｓ’が入力されたらパスワード認証データ更新器１６は、ユーザ保存用の更新されたＰ’とサーバ更新用のパスワード認証データＨ’を生成する。サーバ更新用のパスワード認証データＨ’は、例えばＨ’＝ＨＡＳＨ（Ｓ’‖ＰＷ’‖ＩＤ（Ｕ）‖ＩＤ（Ｓ））ｍｏｄｑにより計算できる。ここで、ＩＤ（Ｕ）とＩＤ（Ｓ）はそれぞれユーザとサーバのＩＤを表す。サーバ更新用のパスワード認証データＨ’はユーザが直接にサーバ２に渡したり、郵便で送付したり、あるいは電話で知らせるなどして、安全に通知する必要がある。ユーザの端末装置１の内部にあるメモリ１２は更新されたＰ’＝（Ｓ’，ＨＡＳＨ）を記憶して保存する。

（３）擬似乱数発生器を利用した場合（その１）、擬似乱数発生器を利用した場合（その２）

次に、図１１を参照して、擬似乱数発生器を利用した場合（その１）、擬似乱数発生器を利用した場合（その２）の端末装置１の更新化処理の動作を説明する。

擬似乱数関数を利用した場合（その１とその２）は、メモリ１２に記憶されたハッシュ関数ＨＡＳＨの代わりに疑似乱数関数ＰＲＮＧを用いる以外は、ハッシュ関数を利用した場合（その１とその２）と同様の動作であるため、ここでは詳細な説明を省略する。

＜サーバの更新化処理＞

（１）ハッシュ関数を利用した場合（その１）、ハッシュ関数を利用した場合（その２）、擬似乱数発生器を利用した場合（その１）、擬似乱数発生器を利用した場合（その２）

初めに、図１２を参照して、ハッシュ関数を利用した場合（その１）、ハッシュ関数を利用した場合（その２）、擬似乱数発生器を利用した場合（その１）、擬似乱数発生器を利用した場合（その２）のサーバ２の更新化処理の動作を説明する。サーバ２は、前述したハッシュ関数を利用した場合（その１とその２）、擬似乱数発生器を利用した場合（その１とその２）に関わらず次のように動作する。

まず、サーバ２に備えたメモリ４１に保存されたユーザＩＤとパスワード認証データＨを読み出す。ユーザの端末装置１から送られたサーバ更新用のパスワード認証データＨ’とメモリ４１から読み出したパスワード認証データＨが入力されたらパスワード認証データ更新器２２は、サーバ保存用のパスワード認証データＨをユーザの端末装置１から送られたＨ’に更新する。サーバの内部にあるメモリ４１は更新されたパスワード認証データＨ＝Ｈ’を記憶して保存する。

＜第２実施例＞

以下、本発明の公開鍵暗号方式を使った実施例について説明する。ただし、本発明は以下の各実施例に限定されるものではなく、例えばこれら実施例の構成要素同士を適宜組み合わせてもよい。

ここで、ＲＳＡ公開鍵暗号方式を使った実施例の説明する前に、いくつかの背景知識及び基礎的な記号について説明しておく。

公開鍵暗号方式では、公開鍵（ＰｕｂＫ）および秘密鍵（ＰｒｉＫ）の対（ＰｕｂＫ，ＰｒｉＫ）が存在する。公開鍵は秘密ではなく、誰でも得ることができる。暗号化は、公開鍵を用いてメッセージｍをＣ＝$Ｅｎｃ_{PubK}$（ｍ）となるような暗号文Ｃを生成することが可能である。暗号文は、秘密鍵を用いてのみ、ｍ＝$Ｄｅｃ_{PriK}$（Ｃ）と復号可能である。暗号文は、公開鍵を用いては

復号できない。そして公開鍵署名方式では、メッセージmを署名化してs＝Sig$_{PrIK}$(m)となるような署名文(m, s)を生成することが可能である。署名文の検証は、公開鍵を用いて、m'＝Ver$_{PubK}$(s)を求め、mとm'を比較することで可能である。つまり、mとm'が一致した場合はその署名文(m, s)が正しいことを検証する。そうではない場合は、(m, s)が正しい署名文ではないことになる。

周知のＲＳＡ公開鍵方式では、公開鍵は(N, e)であり、秘密鍵は(N, d)である。ただし、Nは、２つのランダムに選択された大きな素数pとqの積であり(すなわち、N＝p・q)、eは、eと(p－1)・(q－1)の最大公約数が１であるような小さい任意の数(例えば、e＝３あるいはe＝2^{16}＋1)であり、dは、e^{-1}mod((p－1)・(q－1))である。安全性を最大化するには、pとqを同じ長さにする。メッセージm(m∈Z_N^*)に対して暗号化関数は、Enc$_{PubK}$(m)＝m^emod Nであり、復号化関数は、Dec$_{PrIK}$(C)＝C^dmod Nである。ここで、暗号文Cと公開鍵(N, e)が与えられた時、メッセージmを求めるのは計算量的に困難である。ＲＳＡは、大きな数Nの素因数分解が難しいことから安全性を得ている。そして署名化関数は、Sig$_{PrIK}$(m)＝m^dmod Nであり、検証化関数は、Ver$_{PubK}$(s)＝s^emod Nである。一般に、暗号システムでは、その安全性のレベルを記述する安全性パラメータを有する。ここでは、ハッシュ関数ＨＡＳＨの安全性パラメータとしてkを用い(ただし、１／2^kは無視できるほど小さいと仮定する)、ＲＳＡ公開鍵方式の安全性パラメータとして１を用い、特に、ＲＳＡの法Nは長さｌであると仮定する。また、{0, 1}*は有限の２進数のストリングの集合を、{0, 1}^{k}は長さkの２進数のストリングの集合を示す。ハッシュ関数ＨＡＳＨは{0, 1}*の入力から{0, 1}^{k}の出力を出す安全な一方向関数であり、ＦＤＨ(Ｆｕｌｌ－Ｄｏｍａｉｎ　Ｈａｓｈ)関数は{0, 1}*の入力からZ_N^*\{1}の出力を出す安全な一方向関数である。また、乱数発生器から発生される乱数はT(T∈Z_N^*)を無作為に生成する。また、‖は値を連結(ｃｏｎｃａｔｅｎａｔｉｏｎ)するという意味である。

＜端末装置の初期化＞

ユーザは、サーバに対して個人登録したい時、自分の端末装置の初期化を行う。図１は、ユーザの端末装置の初期化処理の構成を示すブロック図である。初期化は、ユーザがパスワードを入力すると、データ伸張器１１によって、サーバ登録用のパスワード認証データＨと、ユーザ保存用の値Ｐ’が生成され、パスワード認証データＨは、サーバに受け渡され、値Ｐ’は、メモリ１２へ保存する。ここで、データ伸長器１１は、多項式とＦＤＨ関数、ＦＤＨ関数などで構成することが可能である。

（１）多項式とＦＤＨ関数を利用する場合（その１）

初めに、図１３を参照して、多項式とＦＤＨ関数を利用する場合（その１）について説明する。

まず、ＦＤＨ関数発生器１２２によりランダムにＦＤＨ関数ＦＤＨを発生する。そして、多項式発生器１２３によりランダムに多項式を発生する。このとき、登録するサーバの数が一つだったらｘを変数とする１次多項式（ｐ’（ｘ）＝α_1・ｘｍｏｄＮ）を、サーバの数がｎ個だったらｎ次多項式（ｐ’（ｘ）＝α_1・ｘ＋α_2・x^2＋・・・＋α_n・x^nｍｏｄＮ）を発生する。ここで、αはZ_N^*から無作為に選ばれる。例えば、一つのサーバの場合、ｐ’（ｘ）は、ｐ’（ｘ）＝α_1・ｘｍｏｄＮとなる。ここで、ユーザは自分が覚えているパスワード（例えば、”Ｐｏｏｈ９３”）を入力する。多項式とＦＤＨ関数とユーザのパスワードが入力されたらパスワード認証データ生成器１２４は、パスワード認証データＨを生成する。パスワード認証データＨは、例えばＨ＝ｐ（１）＝ｐ’（１）＋Ｐｏｏｈ９３ｍｏｄＮにより計算できる。ここで、ｐ’（１）はｐ’（ｘ）でｘの代わりにサーバのＩＤ（例えば、「１」）を入れて計算した値である。パスワード認証データＨはユーザが直接にサーバに渡したり、郵便で送付したり、あるいは電話で知らせるなどして、安全に通知する必要がある。ユーザの端末装置の内部にあるメモリ１２は多項式発生器から発生された多項式ｐ’（ｘ）とＦＤＨ関数発生器から発生されたＦＤＨ関数ＦＤＨ

とを一緒にP'＝（p'（x），FDH）として記憶して保存する。

（2）多項式とFDH関数を利用する場合（その2）

次に、図13を参照して、多項式とFDH関数を利用する場合（その2）を利用する場合について説明する。

まず、FDH関数発生器122によりランダムにFDH関数FDHを発生する。そして、多項式発生器123によりランダムに多項式を発生する。このとき、登録するサーバの数が一つだったらxを変数とする1次多項式（p'（x）＝α_1・xmodN）を、サーバの数がn個だったらn次多項式（p'（x）＝α_1・x＋α_2・x^2＋・・・＋α_n・x^nmodN）を発生する。ここで、αはZ_N^*から無作為に選ばれる。例えば、一つのサーバの場合、p'（x）は、p'（x）＝α_1・xmodNとなる。ここで、ユーザは自分が覚えているパスワード（例えば、"Pooh93"）を入力する。多項式とFDH関数とユーザのパスワードが入力されたらパスワード認証データ生成器124は、パスワード認証データHを生成する。パスワード認証データHは、例えばH＝p（1）＝p'（1）＋FDH（Pooh93‖ID（U）‖ID（S））modNにより計算できる。ここで、ID（U）とID（S）はそれぞれユーザとサーバのIDを表す。ここで、p'（1）はp'（x）でxの代わりに「1」入れて計算した値である。

例えば、登録するサーバの数がnの場合、パスワード認証データ生成器124は、i番目のサーバに対してパスワード認証データHを生成する。パスワード認証データHは、例えばH＝p（i）＝p'（i）＋FDH（Pooh93‖ID（U）‖ID（S））modNによりに計算できる。ここで、ID（U）とID（S）はそれぞれユーザとi番目のサーバのIDを表す。ここで、p'（i）はn次多項式p'（x）でxの代わりに「i」を入れて計算した値である。

パスワード認証データHはユーザが直接にサーバに渡したり、郵便で送付したり、あるいは電話で知らせるなどして、安全に通知する必要がある。ユーザの端末装置の内部にあるメモリ12は多項式発生器から発生された多項式p'

（ｘ）とＦＤＨ関数発生器から発生されたＦＤＨ関数ＦＤＨとを一緒にＰ’＝（ｐ’（ｘ），ＦＤＨ）として記憶して保存する。

（３）ＦＤＨ関数を利用する場合

次に、図１４を参照して、ＦＤＨ関数を利用する場合について説明する。

まず、ＦＤＨ関数発生器１２５によりランダムにＦＤＨ関数ＦＤＨを発生する。そして、秘密値発生器１２６によりランダムに秘密値Ｓを発生する。ただし、Ｓは全探索攻撃を防げるような長さの値である（例えば、Ｓは８０ビット以上の値）。ここで、ユーザは自分が覚えているパスワード（例えば、”Ｐｏｏｈ９３”）を入力する。ＦＤＨ関数と秘密値Ｓとユーザのパスワードが入力されたらパスワード認証データ生成器１２７はパスワード認証データＨを生成する。パスワード認証データＨは、例えば、Ｈ＝ＦＤＨ（Ｓ‖Ｐｏｏｈ９３‖ＩＤ（Ｕ）‖ＩＤ（Ｓ））により計算できる。ここで、ＩＤ（Ｕ）とＩＤ（Ｓ）はそれぞれユーザとサーバのＩＤを表す。パスワード認証データＨはユーザが直接にサーバに渡したり、郵便で送付したり、あるいは電話で知らせるなどして、安全に通知する必要がある。ユーザの端末装置の内部にあるメモリ１２は秘密値発生器１２６から発生された秘密値ＳとＦＤＨ関数発生器１２５から発生されたＦＤＨ関数ＦＤＨとを一緒にＰ’＝（Ｓ，ＦＤＨ）として記憶して保存する。

＜端末装置とサーバとの初期化＞

サーバは、ＲＳＡ公開鍵をユーザに送信したい時、初期化処理を行う。サーバは、ＲＳＡ公開鍵方式に従って公開鍵と秘密鍵の対を生成し、その公開鍵をユーザに送信する。ここで、初期化は、安全な通信、安全ではない通信などで実現することが可能である。安全ではない通信を利用する場合では、ユーザは、受信した公開鍵が正しいであるかどうかを決定する。このユーザによる決定は、ユーザに、サーバが適当な方法で選択した公開鍵を生成したかどうかを決定する方法を提供する。つまり、サーバが提供した公開鍵ｅと（ｐ－１）・（ｑ－１）の最大公約数が１で（すなわち、ｇｃｄ（ｅ，（ｐ－１）・（ｑ－１））

＝１）あるかどうかを決定するためにＲＳＡ署名方式を使う。

（１）安全な通信を利用する場合

初めに、図１５を参照して、安全な通信を利用する場合の初期化処理について説明する。

まず、ＲＳＡ鍵生成器２３により公開鍵（Ｎ，ｅ）と秘密鍵（Ｎ，ｄ）の対を生成する。ＲＳＡ公開鍵（Ｎ，ｅ）はサーバが直接にユーザに渡したり、郵便で送付したり、あるいは電話で知らせるなどして、安全に通知する必要がある。サーバの内部にあるメモリ４１はＲＳＡ秘密鍵（Ｎ，ｄ）を記憶して保存する。

（２）安全ではない通信を利用する場合

次に、図１６を参照して、安全ではない通信を利用する場合の初期化処理について説明する。

まず、ユーザの端末装置１にある乱数発生器１７によりランダムに乱数R_1（$R_1 \in \{0, 1\}^k$）を発生し、サーバに送信する。一方、サーバ２にあるＲＳＡ鍵生成器２４により公開鍵（Ｎ，ｅ）と秘密鍵（Ｎ，ｄ）の対を生成する。そして、乱数発生器２５によりランダムに乱数R_2（$R_2 \in \{0, 1\}^k$）を発生する。ＲＳＡ署名文生成器２６は、端末装置１から受信したR_1とＲＳＡ鍵生成器２４から生成された秘密鍵（Ｎ，ｄ）と乱数発生器２５から発生させた乱数R_2を入力として、$\{m_j\}_{1 \le j \le n}$の署名$\{s_j\}_{1 \le j \le n}$を生成する（ただし、ｎはｎ≧ｌｏｇ$_e$（ＰＷ･（ｅ－１）／ｅ）になるような整数である。ここで、ＰＷはパスワードを表す。）。署名$\{s_j\}_{1 \le j \le n}$は、$\{s_j = m_j^d \bmod N\}_{1 \le j \le n}$により計算する。ここで、$\{m_j\}_{1 \le j \le n}$はＨＡＳＨ（ｎ‖Ｎ‖ｅ‖ＩＤ（Ｕ）‖ＩＤ（Ｓ）‖$R_1$‖$R_2$）の出力から長さ１のブロックをｎ個に分けて得たものである。ＩＤ（Ｕ）とＩＤ（Ｓ）はそれぞれユーザとサーバのＩＤを表す。サーバ２は、公開鍵（Ｎ，ｅ）と計算して得られた署名文（R_2，$\{s_j\}_{1 \le j \le n}$）を端末装置１に送信する。サーバ２の内部に備えたメモリ４１は、ＲＳＡ鍵生成器２４から生成された秘密鍵（Ｎ，ｄ）を記憶して保存する。

ユーザの端末装置１にある検証結果判断部１８は、乱数発生器１７において発生させたR_1とサーバ２から受信した（（N，e），（R_2，$\{s_j\}_{1\leq j\leq n}$））を入力として、署名文（R_2，$\{s_j\}_{1\leq j\leq n}$）の検証を行う。$\{m_j=s_j^e \bmod N\}_{1\leq j\leq n}$により$\{m_j\}_{1\leq j\leq n}$を計算してこの$\{m_j\}_{1\leq j\leq n}$とHASH（n‖N‖e‖ID（U）‖ID（S）‖$R_1$‖$R_2$）を比較する。検証結果判断部１８において$\{m_j\}_{1\leq j\leq n}$とHASH（n‖N‖e‖ID（U）‖ID（S）‖$R_1$‖$R_2$）が一致しない場合、検証結果判断部１８は、エラー発生器１９に対して、一致しないことを通知する。これを受けて、エラー発生器１９はエラーを発生して処理を中断する。一方、検証結果判断部１８において$\{m_j\}_{1\leq j\leq n}$とHASH（n‖N‖e‖ID（U）‖ID（S）‖R_1‖R_2）が一致した場合はサーバ２の公開鍵（N，e）が正当なものとして検証して、ユーザの端末装置１の内部に備えたメモリ１２に、公開鍵（N，e）を記憶して保存する。

次に、図１７、１８を参照して、前述した初期化を行った端末装置１とサーバ２（図５参照）との間で相互認証及び鍵交換を行う動作を説明する。

＜端末装置の動作＞

（１）多項式とＦＤＨ関数を利用した場合（その１）

初めに、多項式とＦＤＨ関数を利用した場合（その１）の端末装置１の動作を説明する。

まず、ユーザの端末装置１に備えたメモリ１２から記憶された多項式とＦＤＨ関数Ｐ’＝（ｐ’（ｘ），ＦＤＨ）を読み出す。結合器５２はメモリ１２から読み出した多項式ｐ’（ｘ）とＦＤＨ関数ＦＤＨとユーザが入力したパスワードによりＷ＝ＦＤＨ（ｐ（ｘ）‖ＩＤ（Ｕ）‖ＩＤ（Ｓ））を計算して出力する。ここで、ｐ（ｘ）＝ｐ’（ｘ）＋Ｐｏｏｈ９３ｍｏｄＮにより計算する。例えば、ｐ’（ｘ）が１次多項式の場合、ｐ（ｘ）＝ｐ（１）＝ｐ’（１）＋Ｐｏｏｈ９３＝α_1・１＋Ｐｏｏｈ９３ｍｏｄＮにより計算する。ここで、ｐ’（１）はｐ’（ｘ）でｘの代わりに「１」入れて計算した値である。

ユーザの端末装置１に備えたメモリ１２から読み出した多項式ｐ’（ｘ）がｎ次多項式の場合、結合器５２は多項式ｐ’（ｘ）とＦＤＨ関数ＦＤＨとユー

ザが入力したパスワードによりW＝ＦＤＨ（ｐ（ｘ）‖ＩＤ（U）‖ＩＤ（S））を計算して出力する。ここで、ｐ（ｘ）＝ｐ’（ｘ）＋Ｐｏｏｈ９３ｍｏｄＮにより計算する。例えば、ｐ（ｘ）＝ｐ（ｉ）＝ｐ’（ｉ）＋Ｐｏｏｈ９３ｍｏｄＮにより計算する。ここで、ｐ’（ｉ）はｉ番目のサーバに対してｐ’（ｘ）でｘの代わりに「ｉ」入れて計算した値である。

マスク演算器５４は、メモリ１２から読み出した公開鍵（Ｎ，ｅ）と結合器５２から入力されたＷと乱数発生器５３においてランダムに発生させた乱数Ｔ（$T \in Z_N^*$）とからＺを、$Z = T^e \cdot W \bmod N$により計算する。通信処理部５５はＺをサーバ２へ送信し、サーバ２からV_2を受信する。

続いて認証結果判断部５６は、乱数発生器５３から出力されたＴを入力として、ＨＡＳＨ（０１‖Ｔ‖ＩＤ（U）‖ＩＤ（Ｓ））を計算してサーバ２から受信したV_2と比較する。ここで、ＨＡＳＨの代わりにＭＡＣを使ってもよい。認証結果判断部５６においてV_2とＨＡＳＨ（０１‖Ｔ‖ＩＤ（U）‖ＩＤ（S））が一致しない場合、認証結果判断部５６は、エラー発生器５７に対して、一致しないことを通知する。これを受けて、エラー発生器５７はエラーを発生して処理を中断する。一方、認証結果判断部５６においてV_2とＨＡＳＨ（０１‖Ｔ‖ＩＤ（U）‖ＩＤ（Ｓ））が一致した場合はサーバ２が正当な装置として認証して、検証子生成器５８は、V_1＝ＨＡＳＨ（００‖Ｔ‖ＩＤ（U）‖ＩＤ（Ｓ））により検証子V_1を計算してサーバ２へ送信する。同時に、セッション鍵生成器５９は、ＳＫ＝ＨＡＳＨ（１１‖Ｔ‖ＩＤ（U）‖ＩＤ（Ｓ））によりセッション鍵ＳＫを生成する。

（２）多項式とＦＤＨ関数を利用した場合（その２）

次に、多項式とＦＤＨ関数を利用した場合（その２）の端末装置１の動作を説明する。

まず、ユーザの端末装置１に備えたメモリ１２から記憶された多項式とＦＤＨ関数Ｐ’＝（ｐ’（ｘ），ＦＤＨ）を読み出す。結合器５２はメモリ１２から読み出した多項式ｐ’（ｘ）とＦＤＨ関数ＦＤＨとユーザが入力したパスワードによりW＝ＦＤＨ（ｐ（ｘ）‖ＩＤ（U）‖ＩＤ（Ｓ））を計算して出力

する。ここで、ｐ（ｘ）＝ｐ’（ｘ）＋ＦＤＨ（Ｐｏｏｈ９３‖ＩＤ（Ｕ）‖ＩＤ（Ｓ））ｍｏｄＮにより計算する。例えば、ｐ’（ｘ）が１次多項式の場合、ｐ（ｘ）＝ｐ（１）＝ｐ’（１）＋ＦＤＨ（Ｐｏｏｈ９３‖ＩＤ（Ｕ）‖ＩＤ（Ｓ））＝α_1・１＋ＦＤＨ（Ｐｏｏｈ９３‖ＩＤ（Ｕ）‖ＩＤ（Ｓ））ｍｏｄＮにより計算する。ここで、ｐ’（１）はｐ’（ｘ）でｘの代わりに「１」入れて計算した値である。

ユーザの端末装置１に備えたメモリ１２から読み出した多項式ｐ’（ｘ）がｎ次多項式の場合、結合器５２は多項式ｐ’（ｘ）とＦＤＨ関数ＦＤＨとユーザが入力したパスワードによりＷ＝ＦＤＨ（ｐ（ｘ）‖ＩＤ（Ｕ）‖ＩＤ（Ｓ））を計算して出力する。ここで、ｐ（ｘ）＝ｐ’（ｘ）＋ＦＤＨ（Ｐｏｏｈ９３‖ＩＤ（Ｕ）‖ＩＤ（Ｓ））ｍｏｄＮにより計算する。例えば、ｐ（ｘ）＝ｐ（ｉ）＝ｐ’（ｉ）＋ＦＤＨ（Ｐｏｏｈ９３‖ＩＤ（Ｕ）‖ＩＤ（Ｓ））ｍｏｄＮにより計算する。ここで、ｐ’（ｉ）はｉ番目のサーバに対してｐ’（ｘ）でｘの代わりに「ｉ」入れて計算した値である。

マスク演算器５４は、メモリ１２から読み出した公開鍵（Ｎ，ｅ）と結合器５２から入力されたＷと乱数発生器５３においてランダムに発生させた乱数Ｔ（$T \in Z_N^*$）とからＺを、$Z = T^e \cdot W \bmod N$により計算する。通信処理部５５はＺをサーバ２へ送信し、サーバ２からV_2を受信する。

続いて認証結果判断部５６は、乱数発生器５３から出力されたＴを入力として、ＨＡＳＨ（０１‖Ｔ‖ＩＤ（Ｕ）‖ＩＤ（Ｓ））を計算してサーバ２から受信したV_2と比較する。ここで、ＨＡＳＨの代わりにＭＡＣを使ってもよい。認証結果判断部５６においてV_2とＨＡＳＨ（０１‖Ｔ‖ＩＤ（Ｕ）‖ＩＤ（Ｓ））が一致しない場合、認証結果判断部５６は、エラー発生器５７に対して、一致しないことを通知する。これを受けて、エラー発生器５７はエラーを発生して処理を中断する。一方、認証結果判断部５６においてV_2とＨＡＳＨ（０１‖Ｔ‖ＩＤ（Ｕ）‖ＩＤ（Ｓ））が一致した場合はサーバ２が正当な装置として認証して、検証子生成器５８は、V_1＝ＨＡＳＨ（００‖Ｔ‖ＩＤ（Ｕ）‖ＩＤ（Ｓ））により検証子V_1を計算してサーバ２へ送信する。同時に、セッション鍵生成器５９は、ＳＫ＝ＨＡＳＨ（１１‖Ｔ‖ＩＤ（Ｕ）‖ＩＤ（Ｓ））

によりセッション鍵ＳＫを生成する。

（３）ＦＤＨ関数を利用した場合

次に、ＦＤＨ関数を利用した場合の端末装置１の動作を説明する。

まず、ユーザの端末装置１に備えたメモリ１２から記憶された秘密値とＦＤＨ関数Ｐ’＝（Ｓ，ＦＤＨ）を読み出す。結合器５２はメモリ１２から読み出した秘密値ＳとＦＤＨ関数ＦＤＨとユーザが入力したパスワードによりＷを計算して出力する。例えば、Ｗは、Ｗ＝ＦＤＨ（Ｓ‖Ｐｏｏｈ９３‖ＩＤ（Ｕ）‖ＩＤ（Ｓ））により計算する。マスク演算器５４は、メモリ１２から読み出した公開鍵（Ｎ，ｅ）と結合器５２から入力されたＷと乱数発生器５３においてランダムに発生させた乱数Ｔ（$T \in Z_N^*$）とからＺを、$Z = T^e \cdot W \bmod N$により計算する。通信処理部５５はＺをサーバ２へ送信し、サーバ２からV_2を受信する。

続いて認証結果判断部５６は、乱数発生器５３から出力されたＴを入力として、ＨＡＳＨ（０１‖Ｔ‖ＩＤ（Ｕ）‖ＩＤ（Ｓ））を計算してサーバ２から受信したV_2と比較する。ここで、ＨＡＳＨの代わりにＭＡＣを使ってもよい。認証結果判断部５６においてV_2とＨＡＳＨ（０１‖Ｔ‖ＩＤ（Ｕ）‖ＩＤ（Ｓ））が一致しない場合、認証結果判断部５６は、エラー発生器５７に対して、一致しないことを通知する。これを受けて、エラー発生器５７はエラーを発生して処理を中断する。一方、認証結果判断部５６においてV_2とＨＡＳＨ（０１‖Ｔ‖ＩＤ（Ｕ）‖ＩＤ（Ｓ））が一致した場合はサーバ２が正当な装置として認証して、検証子生成器５８は、V_1＝ＨＡＳＨ（００‖Ｔ‖ＩＤ（Ｕ）‖ＩＤ（Ｓ））により検証子V_1を計算してサーバ２へ送信する。同時に、セッション鍵生成器５９は、ＳＫ＝ＨＡＳＨ（１１‖Ｔ‖ＩＤ（Ｕ）‖ＩＤ（Ｓ））によりセッション鍵ＳＫを生成する。

＜サーバの動作＞

（１）多項式とＦＤＨ関数を利用した場合（その１）、多項式とＦＤＨ関数を利用した場合（その２）

サーバ２は、前述した多項式とＦＤＨ関数を利用した場合（その１）、多項式とＦＤＨ関数を利用した場合（その２）に関わらず次のように動作する。

まず、サーバ２に備えたメモリ４１から保存されたユーザＩＤとパスワードの認証データＨを読み出す。マスター鍵生成器６２はメモリ４１から読み出したＨと秘密鍵（Ｎ，ｄ）と端末装置１から受信したＺを入力としてＴを、Ｔ＝（Ｚ／Ｗ）^{d}ｍｏｄＮにより計算して出力する。ここで、Ｗは、Ｗ＝ＦＤＨ（Ｈ‖ＩＤ（Ｕ）‖ＩＤ（Ｓ））により計算する。検証子生成器６３は、マスター鍵生成器６２から入力されたＴから検証子V_2を、V_2＝ＨＡＳＨ（０１‖Ｔ‖ＩＤ（Ｕ）‖ＩＤ（Ｓ））により計算する。通信処理部６４は、計算して得られたV_2を端末装置１へ送信し、端末装置１から受信したV_1を認証結果判断部６５へ出力する。

続いて認証結果判断部６５は、マスター鍵生成器６２から出力されたＴを入力として、ＨＡＳＨ（００‖Ｔ‖ＩＤ（Ｕ）‖ＩＤ（Ｓ））を計算して端末装置１から受信したV_1と比較する。ここで、ＨＡＳＨの代わりにＭＡＣを使ってもよい。認証結果判断部６５においてV_1とＨＡＳＨ（００‖Ｔ‖ＩＤ（Ｕ）‖ＩＤ（Ｓ））が一致しない場合、認証結果判断部６５は、エラー発生器６６に対して、一致しないことを通知する。これを受けて、エラー発生器６６はエラーを発生して処理を中断する。一方、認証結果判断部６５においてV_1とＨＡＳＨ（００‖Ｔ‖ＩＤ（Ｕ）‖ＩＤ（Ｓ））が一致した場合は端末装置１が正当な装置として認証して、セッション鍵生成器６７は、ＳＫ＝ＨＡＳＨ（１１‖Ｔ‖ＩＤ（Ｕ）‖ＩＤ（Ｓ））によりセッション鍵ＳＫを生成する。

（２）ＦＤＨ関数を利用した場合

次に、ＦＤＨ関数を利用した場合のサーバ２の動作を説明する。

まず、サーバ２に備えたメモリ４１から保存されたユーザＩＤとパスワードの認証データＨを読み出す。マスター鍵生成器６２はメモリ４１から読み出したＨと秘密鍵（Ｎ，ｄ）と端末装置１から受信したＺを入力としてＴを、Ｔ＝（Ｚ／Ｗ）^{d}ｍｏｄＮにより計算して出力する。ここで、Ｗは、Ｗ＝Ｈである。検証子生成器６３は、マスター鍵生成器６２から入力されたＴから検証子V_2

を、V_2＝ＨＡＳＨ（０１‖Ｔ‖ＩＤ（Ｕ）‖ＩＤ（Ｓ））により計算する。通信処理部６４は、計算して得られたV_2を端末装置１へ送信し、端末装置１から受信したV_1を認証結果判断部６５へ出力する。

続いて認証結果判断部６５は、マスター鍵生成器６２から出力されたＴを入力として、ＨＡＳＨ（００‖Ｔ‖ＩＤ（Ｕ）‖ＩＤ（Ｓ））を計算して端末装置１から受信したV_1と比較する。ここで、ＨＡＳＨの代わりにＭＡＣを使ってもよい。認証結果判断部６５においてV_1とＨＡＳＨ（００‖Ｔ‖ＩＤ（Ｕ）‖ＩＤ（Ｓ））が一致しない場合、認証結果判断部６５は、エラー発生器６６に対して、一致しないことを通知する。これを受けて、エラー発生器６６はエラーを発生して処理を中断する。一方、認証結果判断部６５においてV_1とＨＡＳＨ（００‖Ｔ‖ＩＤ（Ｕ）‖ＩＤ（Ｓ））が一致した場合は端末装置１が正当な装置として認証して、セッション鍵生成器６７は、ＳＫ＝ＨＡＳＨ（１１‖Ｔ‖ＩＤ（Ｕ）‖ＩＤ（Ｓ））によりセッション鍵ＳＫを生成する。

＜パスワード認証データの更新－１＞

ユーザは、サーバに対してすでに登録されたパスワード認証データを、自分が覚えているパスワードを変えずに、更新したい時、自分の端末装置の更新化を行う。図９、１９は、ユーザの端末装置の更新化処理の構成を示すブロック図である。ここでの更新化処理は、前述した多項式とＦＤＨ関数を利用した場合（その１）、多項式とＦＤＨ関数を利用した場合（その２）とマスター鍵を利用する場合に適用することが可能である。また、この更新化処理によりサーバに対するｒｅｐｌａｙ攻撃を防ぐことが可能である。

＜端末装置の更新化処理＞

（１）多項式とＦＤＨ関数を利用した場合（その１）、多項式とＦＤＨ関数を利用した場合（その２）

初めに、図９を参照して、多項式とＦＤＨ関数を利用した場合（その１）、多項式とＦＤＨ関数を利用した場合（その２）の端末装置１の更新化処理を説明する。端末装置１は、前述した多項式とＦＤＨ関数を利用した場合（その１）、

多項式とＦＤＨ関数を利用した場合（その２）に関わらず次のように動作する。

まず、多項式発生器１３によりランダムに多項式を発生する。このとき、登録したサーバの数が一つだったらｘを変数とする１次多項式（ｔ’（ｘ）＝β_1・ｘｍｏｄＮ）を、サーバの数がｎ個だったらｎ次多項式（ｔ’（ｘ）＝β_1・ｘ＋β_2・x^2＋・・・＋β_n・x^nｍｏｄＮ）を発生する。ここで、βはZ_N^*から無作為に選ばれる。例えば、一つのサーバの場合、Ｔ’＝ｔ’（ｘ）は、ｔ’（ｘ）＝β_1・ｘｍｏｄＮとなる。ここで、ユーザの端末装置１に備えたメモリ１２から記憶された多項式とＦＤＨ関数Ｐ’＝（ｐ’（ｘ），ＦＤＨ）を読み出す。多項式ｔ’（ｘ）と多項式ｐ’（ｘ）が入力されたら更新値生成器１４は、ユーザ保存用の更新された多項式Ｐ’とサーバ更新用の値Ｈ’を生成する。更新された多項式Ｐ’は、例えばＰ’＝ｔ’（ｘ）＋ｐ’（ｘ）＝（α_1＋β_1）・ｘｍｏｄＮにより計算できる。サーバ更新用の値Ｈ’、例えばＨ’＝ｔ’（１）ｍｏｄＮにより計算できる。ここで、ｔ’（１）はｔ’（ｘ）でｘの代わりに「１」を入れて計算した値である。

例えば、登録したサーバの数がｎの場合、更新値生成器１４は、ｉ番目のサーバに対してサーバ更新用の値Ｈ’を生成する。サーバ更新用の値Ｈ’は、例えばＨ’＝ｔ’（ｉ）ｍｏｄＮにより計算できる。ここで、ｔ’（ｉ）はｎ次多項式ｔ’（ｘ）でｘの代わりに「ｉ」を入れて計算した値である。

サーバ更新用の値Ｈ’はユーザが直接にサーバに渡したり、郵便で送付したり、あるいは電話で知らせるなどして、安全に通知する必要がある。ユーザの端末装置の内部にあるメモリ１２は更新された多項式Ｐ’＝ｔ’（ｘ）＋ｐ’（ｘ）とメモリ１２から読み出したＦＤＨ関数ＦＤＨとを一緒にＰ’＝（ｔ’（ｘ）＋ｐ’（ｘ），ＦＤＨ）として記憶して保存する。

（２）マスター鍵を利用する場合

次に、図１９を参照して、マスター鍵を利用する場合の端末装置１の更新化処理の動作を説明する。

まず、乱数発生器５３によりランダムに乱数Ｔ（Ｔ∈Z_N^*）を発生する。ここで、ユーザの端末装置１に備えたメモリ１２から記憶された多項式とＦＤＨ

関数Ｐ’＝（ｐ’（ｘ），ＦＤＨ）を読み出す。乱数Ｔと多項式ｐ’（ｘ）が入力されたら更新値生成器２０は、ユーザ保存用の更新された多項式Ｐ’を生成する。更新された多項式Ｐ’は、Ｐ’＝Ｔ＋ｐ’（ｘ）ｍｏｄＮにより計算できる。ユーザの端末装置の内部にあるメモリ１２は更新された多項式Ｐ’＝Ｔ＋ｐ’（ｘ）とメモリ１２から読み出したＦＤＨ関数ＦＤＨとを一緒にＰ’＝（Ｔ＋ｐ’（ｘ），ＦＤＨ）として記憶して保存する。

＜サーバの更新化処理＞

（１）多項式とＦＤＨ関数を利用した場合（その１）、多項式とＦＤＨ関数を利用した場合（その２）

初めに、図１０を参照して、多項式とＦＤＨ関数を利用した場合（その１）、多項式とＦＤＨ関数を利用した場合（その２）のサーバ２の更新化処理の動作を説明する。サーバ２は、前述した多項式とＦＤＨ関数を利用した場合（その１）、多項式とＦＤＨ関数を利用した場合（その２）に関わらず次のように動作する。

まず、サーバ２に備えたメモリ４１に保存されたユーザＩＤとパスワード認証データＨを読み出す。ユーザの端末装置から送られたサーバ更新用の値Ｈ’とメモリ４１から読み出したパスワード認証データＨが入力されたら更新値生成器２１は、サーバ保存用の更新されたパスワード認証データＨを生成する。更新されたパスワード認証データＨは、例えばＨ＝ｐ（１）＋ｔ’（１）ｍｏｄＮにより計算できる。サーバの内部にあるメモリ４１は更新されたパスワード認証データＨを記憶して保存する。

（２）マスター鍵を利用する場合

次に、図２０を参照して、マスター鍵を利用する場合のサーバ２の更新化処理の動作を説明する。

まず、マスター鍵生成器６２によりマスター鍵Ｔを生成する。ここで、サーバ２に備えたメモリ４１から保存されたユーザＩＤとパスワード認証データＨを読み出す。マスター鍵Ｔとメモリ４１から読み出したパスワード認証デー

タHが入力されたら更新値生成器２７は、サーバ保存用の更新されたパスワード認証データHを生成する。更新されたパスワード認証データHは、H＝ｐ（１）＋TmodNにより計算できる。サーバの内部にあるメモリ４１は更新されたパスワード認証データHを記憶して保存する。

＜パスワード認証データの更新－２＞

ユーザは、サーバに対してすでに登録されたパスワード認証データを、自分が覚えているパスワードを変えながら、更新したい時、自分の端末装置の更新化処理を行う。図１１は、ユーザの端末装置の更新化処理の構成を示すブロック図である。更新化処理は、秘密値発生器１５による秘密値S'とユーザの新しいパスワードPW'とユーザの端末装置１に備えたメモリ１２から記憶されたP'が入力されると、パスワード認証データ更新器１６によって、サーバ更新用のパスワード認証データH'と、ユーザ保存用の更新されたP'が生成され、H'は、サーバ２に受け渡され、更新されたP'は、メモリ１２へ保存する。ここでの更新化処理は、前述したＦＤＨ関数を利用した場合に適用することが可能である。また、この更新化処理によりサーバに対するｒｅｐｌａｙ攻撃を防ぐことが可能である。そして、多項式とＦＤＨ関数を利用した場合（その１），多項式とＦＤH関数を利用した場合（その２）も同じように初期化処理と同様の動作を用いて適用することが可能なため、ここでは詳細な説明を省略する。

＜端末装置の更新化処理＞

（１）ＦＤＨ関数を利用した場合

図１１を参照して、ＦＤＨ関数を利用した場合の端末装置１の更新化処理の動作を説明する。

まず、秘密値発生器１５によりランダムに秘密値S'を発生する。ここで、ユーザの端末装置１に備えたメモリ１２から記憶されたP'＝（S，ＦＤH）を読み出す。ユーザが覚えている新しいパスワード（PW'）とＦＤＨ関数ＦＤＨと秘密値S'が入力されたらパスワード認証データ更新器１６は、ユーザ保存用の更新されたP'とサーバ更新用のパスワード認証データH'を生成する。サーバ

更新用のパスワード認証データH'は、例えばH'＝ＦＤＨ（S'‖ＰW'‖ＩＤ（U）‖ＩＤ（S））により計算できる。ここで、ＩＤ（U）とＩＤ（S）はそれぞれユーザとサーバのＩＤを表す。サーバ更新用のパスワード認証データH'はユーザが直接にサーバ２に渡したり、郵便で送付したり、あるいは電話で知らせるなどして、安全に通知する必要がある。ユーザの端末装置１の内部にあるメモリ１２は更新されたP'＝（S', ＦＤＨ）を記憶して保存する。

＜サーバの更新化処理＞

（1）ＦＤＨ関数を利用した場合

図１２を参照して、ＦＤＨ関数を利用した場合のサーバ２の更新化処理の動作を説明する。

まず、サーバ２に備えたメモリ４１に保存されたユーザＩＤとパスワード認証データＨを読み出す。ユーザの端末装置１から送られたサーバ更新用のパスワード認証データH'とメモリ４１から読み出したパスワード認証データＨが入力されたらパスワード認証データ更新器２２は、サーバ保存用のパスワード認証データＨをユーザの端末装置１から送られたH'に更新する。サーバの内部にあるメモリ４１は更新されたパスワード認証データＨ＝H'を記憶して保存する。

このように、多項式を利用することにより、不正利用しようと思っている者が他人の端末装置を持っていたとしてもユーザのパスワードは情報理論的に安全である。また、サーバ内に侵入して保存されている情報を得たとしてもユーザのパスワードは情報理論的に安全である。また、ハッシュ関数と擬似乱数発生器とＦＤＨ関数を利用する場合には、不正利用しようと思っている者にとってユーザのパスワードは計算量的に安全である。

次に、前述した技術を応用した装置について説明する。

＜端末に分散データを保存しない場合の遠隔分散保存装置におけるデータ保存処理＞

次に、図２１を参照して、端末に分散データを保存しない場合の遠隔分散保存装置におけるデータ保存処理について説明する。図２１は、端末に分散データを保存しない場合の遠隔分散保存装置５の構成を示すブロック図である。

ユーザは分散保存したいデータＤＡＴＡを自分の端末装置２１で処理し、ｎ台のサーバに保存するデータＳ’１，…，Ｓ’ｎへと分割する。分割データＳ’ｉはデータのＩＤであるＤＩＤと共にユーザの端末装置２１によりサーバの認証装置と共有された鍵ＳＫｉを用いて作成された安全な通信路を使用可能な通信器５２によってサーバＩＤｉへと送られ保存される。同様の方法により、保存データの一覧情報を分割し、サーバに保存することも可能である。また、ユーザの端末装置（認証データ更新モード）を適当な間隔（記憶情報がオフライン全数探索で求まる間隔より短い間隔で、例えば、認証を行う度や２，３日に１回など）で動作させ、Ｐ’と各サーバに記録されているＨを更新する情報ＵＰ’，ＵＨ１，…，ＵＨｎを生成させ、それらを更新する。

これにより、サーバに保存されるデータと認証用データをともに、漏洩と破損に強くなるように構成することができる。漏洩と破損への耐性は（ｎ，ＤＳ，ＬＳ１，ＬＳ２）の４組のパラメータで表現できる。ＤＳ，ＬＳ１，ＬＳ２はともに（漏洩や破損の起こる）エンティティの組み合わせの集合であり、ＤＳはデータの破損に対する耐性、ＬＳ１，ＬＳ２は漏洩に対する耐性を表現している。ＤＳには、壊れても構わないエンティティの組み合わせが記述される。災害などの理由によりローカルバックアップを含め保存しているデータが完全に利用できなくなったとしても、利用者が自分のデータを復元できる範囲の破損組み合わせが記述される。ＬＳ１には、記録情報が漏れても構わないエンティティの組み合わせが記述される。保存データが漏洩したとしても、攻撃者が利用者のデータを復元することが困難な範囲の漏洩組み合わせとする。ＬＳ２には、記録情報が漏れたとしても、後で何らかの対抗策がとれる範囲のエンティティの組み合わせが記述される。保存データが漏洩したとしても、攻撃者が利用者のデータを復元することを困難にする対抗策の存在する範囲の漏洩組み合わせとする。

また、ユーザの記憶情報はオフラインで全数探索できる程少量であると仮定

すると、従来の漏洩に弱い認証方式を使う場合、攻撃者は漏洩情報と通信路上で得た情報を使って利用者の記憶情報を全数探索でき、結果として、利用者に成りすまして遠隔分散保存されているデータをすべて入手できる。つまり、ＬＳ１にサーバ｛Ｓ｝とユーザの所有物｛Ｕ｝をそれぞれ含めることはできなかった。これに対して、漏洩に強い認証方式を使うことでＬＳ１にサーバ｛Ｓ｝とユーザの所有物｛Ｕ｝をそれぞれ含めることが可能となる。漏洩に強い認証方式を含む全ての遠隔認証方式はユーザの所有物｛Ｕ｝とサーバ｛Ｓ｝の両方から情報が漏洩すると利用者の記憶情報を全数探索でき、結果として、利用者に成りすまして遠隔分散保存されているデータをすべて入手できる。よって、ＬＳ２にユーザの所有物｛Ｕ｝とサーバ｛Ｓ｝の組み合わせ｛ＵＳ｝を入れることはできなかった。これに対して、Ｐ'とＨ１，…，Ｈｎを更新することでＬＳ２にユーザの所有物とサーバの組み合わせを入れることが可能となる。

次に、図２２を参照して、図２１に示すデータ分散器５１の構成を説明する。調整器５１１は入力されたパラメータｎ，ｋを秘密分散器５１２に渡す。秘密分散器５１２は入力されたパラメータｎ，ｋに従い保存データＤＡＴＡを（ｋ，ｎ）分散データＳ１，Ｓ２，…，Ｓｎに変換する。次に、調整器５１１はデータのＩＤであるＤＩＤからデータ伸長器５１３の入力ｘを生成しデータ伸長器５１３へ渡す。データ伸長器５１３は対応する情報Ｈを出力し、それを暗号器５１４に渡す。ここでのＨはオフラインでの全数探索に強い長さのものを利用する。データ伸長器５１３が短いＨを出力する場合には異なるｘを複数データ伸長器５１３に渡し、得られた複数のＨを利用する。暗号器５１４は、Ｈを鍵としてｎ－ｋ＋１個以上の分散データを暗号化する。また、Ｓ１，…，Ｓｎ－ｋ＋１のそれぞれに改ざん検出符号を付けてもよい。暗号器の出力がＳ'１，…，Ｓ'ｎであり、ＤＩＤ、ＩＤ１，ＩＤ２，…，ＩＤｎと共にデータ分割器５１の出力となる。

なお、（ｋ，ｎ）分散データとは、元データがｎ個に分割されたデータであり、かつ、そのどのｋ個からでも元のデータを復元できるがｋ未満からでは復元できないような性質を有するデータである。（ｋ，ｎ）分散データ以外にも

任意のアクセス構造を持った分散データを利用することが可能である。また、秘密分散器は多項式や行列を使った情報量的に安全な分散方式以外に、暗号化を使うことで保存データサイズを少なくできる計算量的に安全な分散方式を用いることが可能である。

このデータ分割器５１は、（ｎ，ＤＳ，ＬＳ１，ＬＳ２）＝（ｎ，｛ＣＳｎ－ｋ｝，｛ＵＣ，ＣＳｎ｝，｛ＵＣＳｋ－１｝）を実現している。｛ＣＳ｝はクライアントおよびサーバの記録情報およびその部分情報を意味し、｛Ｓｎ｝はｎ台のサーバの全記録情報およびその部分情報を意味する。｛Ｃ，Ｓ｝は「クライアントに記録されている全情報およびその部分情報あるいはサーバに記録されている全情報およびその部分情報」を意味する。｛ＵＣＳｋ－１｝からの漏洩は認証情報のアップデート処理で対処できる。攻撃者が｛ＵＣＳｋ－１｝から記憶情報を求める間に認証情報がアップデートされれば攻撃者は利用者のデータを求めることができなくなる。ユーザの所有物｛Ｕ｝の紛失（Ｐ’の紛失）に対しては、｛Ｕ｝に記録してあるデータをアップデートするたびにそのコピーをローカルに作成することで解決できる。災害などでローカルコピーもろとも｛Ｕ｝が破損する危険性に対しては、｛Ｕ｝に記録してあるデータを秘密分散器で（ｋ’，ｎ）分散データで分割して各サーバにも保存しておくことで解決できる。ｋ’＞＝ｋの場合、（ｎ，ＤＳ，ＬＳ１，ＬＳ２）＝（ｎ，｛ＵＣＳｎ－ｋ’，ＣＳｎ－ｋ｝，｛ＵＣ，ＣＳｋ’－１｝，｛ＵＣＳｋ－１｝）を実現しており、ｋ’＜ｋの場合、（ｎ，ＤＳ，ＬＳ１，ＬＳ２）＝（ｎ，｛ＵＣＳｎ－ｋ，ＣＳｎ－ｋ｝，｛ＵＣ，ＣＳｋ’－１｝，｛ＵＣＳｋ－１，ＣＳｋ－１｝）を実現している。

さらに、ユーザの所有物に記録してあるデータと記憶情報のすべて、あるいはそれらの一部も秘密分散器で（ｋ’，ｎ）分散データに分割して各サーバにも保存しておけば、ユーザが仮に記憶情報を忘れたとしてもオフライン解析により（所有物の記録情報と）記憶情報を復元できる。その際、１）すべてを分散する場合はオフライン解析を省くことができ、２）一部を分散する場合はその量に応じてオフライン解析の計算量を削減することができる。この機能により、ユーザは第三者にデータを復号する権限を与える際に、データを復元する

容易さ（攻撃者が｛ＣＳｋ'｝を得た場合のデータ復元の容易さとも同じ）を調整できるようになる。

＜端末にデータを保存しない場合の遠隔分散保存装置におけるデータ復元処理＞

次に、図２３を参照して、端末にデータを保存しない場合の遠隔分散保存装置におけるデータ復元処理について説明する。図２３は、端末にデータを保存しない場合の遠隔分散保存装置５の構成を示すブロック図である。

データ復元器５４は、入力されたＤＩＤに対応する分散データＳ'１，…，Ｓ'ｎの内少なくともｋ個を通信器５２を介してサーバＩＤ１，ＩＤ２，…，ＩＤｎから受け取る。データ復元器５４はＳ'１，…，Ｓ'ｎの内少なくともｋ個を処理してＤＡＴＡを復元する。同様の方法で、保存データの一覧を復元することも可能である。また、ユーザの端末装置２１（認証データ更新モード）を適当な間隔（記憶情報がオフライン全数探索で求まる間隔より短い間隔で、例えば、認証を行う度や２，３日に１回など）で動作させ、Ｐ'と各サーバに記録されているＨを更新する情報ＵＰ'，ＵＨ１，…，ＵＨｎを生成させ、それらを更新する。

次に、図２４を参照して、図２３に示すデータ復元器５４の構成を説明する。調整器５４１は、入力されたｎ台のサーバＩＤ、ＩＤ１，ＩＤ２，…，ＩＤｎとＤＩＤを出力する。また、調整器５４１はＤＩＤからデータ伸長器５４２の入力ｘを生成しデータ伸長器５４２へ渡す。データ伸長器５４２は対応する情報Ｈを出力し、それを暗号器５４３に渡す。暗号器５４３は得られた分散データＳ'１，Ｓ'２，…，Ｓ'ｎの内、暗号化されているものを復号しＳ１，Ｓ２，…，Ｓｎを秘密分散復元器に渡す。秘密分散復元器５４４は渡された分散データからＤＡＴＡを復元する。なお、改ざん検出を行い改ざんの行われていない分散データのみをｋ個渡してもよい。

＜端末にも分散データを保存する場合の遠隔分散保存装置におけるデータ保

存処理＞

次に、図２５を参照して、端末にも分散データを保存する場合の遠隔分散保存装置におけるデータ保存処理について説明する。図２５は、端末にも分散データを保存する場合の遠隔分散保存装置５の構成を示すブロック図である。ここでは、図２１に示す構成を異なる部分のみを説明する。

ユーザは分散保存したいデータＤＡＴＡを自分の端末装置２１で処理し、手元に残すデータＤＬとｎ台のサーバに保存するデータＲＳ’１，…，ＲＳ’ｎへと分割する。ＤＬは手元の記録装置５５に保存され、分割データＲＳ’１はデータのＩＤであるＤＩＤと共にユーザの端末装置２１によりサーバの認証装置と共有された鍵ＳＫｉを用いて作成された安全な通信路を通ってサーバＩＤｉへと送られ保存される。同様の方法で、保存データの一覧情報を分割し、サーバに保存することも可能である。

このように、分散データの一部を自分の端末装置２１に置くことで、サーバとの通信量を削減できる。破損したサーバの組み合わせに応じて通信量が変わる方式と変わらない方式が可能であるが、変わる方法の場合ｎ台のサーバ全体の記憶領域を削減できる。なお、漏洩と破損に対する耐性は、データを手元に置かない場合と同等に保つことができる。

次に、図２６を参照して、図２５に示すデータ分割器５１の構成を説明する。ここでは、図２２に示す構成と異なる部分のみを説明する。暗号器５１５は乱数生成器５１６から乱数Ｒを受け取り、これを鍵として保存データを暗号化し、それをＤＬとして出力する。調整器５１１は入力されたパラメータｎ，ｋを秘密分散器５１２に渡す。秘密分散器５１２は入力されたパラメータｎ，ｋに従いＲを（ｋ，ｎ）分散データＲＳ１，ＲＳ２，…，ＲＳｎに変換する。次に、調整器５１１はデータのＩＤであるＤＩＤからデータ伸長器５１３の入力ｘを生成しデータ伸長器５１３へ渡す。データ伸長器５１３は対応する情報Ｈを出力し、これを鍵として暗号器５１４に渡す。ここでのＨはオフラインでの全数探索に強い長さのものを利用する。データ伸長器５１３が短いＨを出力する場合には異なるｘを複数のデータ伸長器５１３に渡し、得られた複数のＨを利

用する。暗号器５１４は、Ｈを鍵としてｎ－ｋ＋１個以上の分散データを暗号化する。また、ＲＳ１，…，ＲＳｎ－ｋ＋１のそれぞれに改ざん検出符号を付けてもよい。暗号器の出力がＲＳ’１，…，ＲＳ’ｎである。

なお、（ｋ，ｎ）分散データ以外にも任意のアクセス構造を持った分散データを利用することが可能である。また、秘密分散器５１２には多項式や行列を使った情報量的に安全な分散方式や、暗号を使う計算量的に安全な分散方式を用いることが可能であるが、Ｒのサイズが小さい場合には、計算量的に安全な分散方式を用いることによるサイズの削減効果は小さいため、情報量的に安全な分散方式を用いた方がよい。

＜端末にも分散データを保存する場合の遠隔分散保存装置におけるデータ復元処理＞

次に、図２７を参照して、端末にも分散データを保存する場合の遠隔分散保存装置におけるデータ復元処理について説明する。図２７は、端末にも分散データを保存する場合の遠隔分散保存装置５の構成を示すブロック図である。ここでは、図２３に示す構成と異なる部分のみを説明する。

データ復元器５４は入力されたＤＩＤに対応する分散データＲＳ’１，…，ＲＳ’ｎの内少なくともｋ個を通信器を介してサーバＩＤ１，ＩＤ２，…，ＩＤｎから受け取る。データ復元器５４はＲＳ’１，…，ＲＳ’ｎの内少なくともｋ個を処理してＤＡＴＡを復元する。同様の方法で、保存データの一覧を復元することも可能である。

次に、図２８を参照して、図２７に示すデータ復元器５４の構成を説明する。ここでは、図２４に示す構成と異なる部分のみを説明する。調整器５４１は、ＤＩＤからデータ伸長器５４２の入力ｘを生成しデータ伸長器５４２へ渡す。データ伸長器５４２は対応する情報Ｈを出力し、それを復号器５４３に渡す。復号器５４３は得られた分散データＲＳ’１，ＲＳ’２，…，ＲＳ’ｎの内、暗号化されているものを復号しＲＳ１，…，ＲＳｎを秘密分散復元器５４４に渡す。秘密分散復元器５４４は渡された分散データからＤＡＴＡを復号器５４５

によって復元する。また、改ざん検出を行い改ざんの行われていない分散データをｋ個渡してもよい。

なお、図面に示す各処理部の機能を実現するためのプログラムをコンピュータ読み取り可能な記録媒体に記録して、この記録媒体に記録されたプログラムをコンピュータシステムに読み込ませ、実行することにより認証処理、鍵交換処理を行ってもよい。なお、ここでいう「コンピュータシステム」とは、ＯＳや周辺機器等のハードウェアを含むものとする。また、「コンピュータシステム」は、ホームページ提供環境（あるいは表示環境）を備えたWWWシステムも含むものとする。また、「コンピュータ読み取り可能な記録媒体」とは、フレキシブルディスク、光磁気ディスク、ＲＯＭ、ＣＤ－ＲＯＭ等の可搬媒体、コンピュータシステムに内蔵されるハードディスク等の記憶装置のことをいう。さらに「コンピュータ読み取り可能な記録媒体」とは、インターネット等のネットワークや電話回線等の通信回線を介してプログラムが送信された場合のサーバやクライアントとなるコンピュータシステム内部の揮発性メモリ（ＲＡＭ）のように、一定時間プログラムを保持しているものも含むものとする。

また、上記プログラムは、このプログラムを記憶装置等に格納したコンピュータシステムから、伝送媒体を介して、あるいは、伝送媒体中の伝送波により他のコンピュータシステムに伝送されてもよい。ここで、プログラムを伝送する「伝送媒体」は、インターネット等のネットワーク（通信網）や電話回線等の通信回線（通信線）のように情報を伝送する機能を有する媒体のことをいう。また、上記プログラムは、前述した機能の一部を実現するためのものであっても良い。さらに、前述した機能をコンピュータシステムにすでに記録されているプログラムとの組み合わせで実現できるもの、いわゆる差分ファイル（差分プログラム）であっても良い。

産業上の利用の可能性

この発明によれば、端末装置側あるいはサーバ側から装置内に保存している情報が漏れたとしてもオフライン解析によってパスワードを見つけだすことができないため、サーバの不正利用を防止することが可能になるという効果が得られる。また、装置内に保存されている情報を盗まれないようにするための耐タンパー性のモジュールを使用する必要がないため、装置構成を簡単にすることができる。また、公開鍵暗号システムのように複雑な鍵管理処理を行う必要がないため、計算処理を向上させることができるとともに、処理内容を簡単にすることができる。また、複数のサーバに対しても拡張することができる。

さらに、各サーバと端末装置とでユーザＩＤを同期させながら動的に変化させることで、盗聴者がユーザＩＤを用いてユーザのプライバシ情報を紐付けすることを防止することができる。

請 求 の 範 囲

1． 端末装置とサーバ間において相互に認証を行う認証システムであって、

前記端末装置は、

ユーザ保存用の認証情報Ｐ’を予め記憶しておく記憶手段と、

前記記憶手段から読み出した認証情報Ｐ’と認証時に入力されたパスワードを入力として所定の計算式により値Ｐを求める結合手段と、

前記値Ｐと内部において発生させた乱数を入力として所定の計算式により値Ｙ１を求め、前記サーバへ送信するマスク演算手段と、

前記値Ｐと内部において発生させた乱数と前記サーバから受信した値Ｙ２を入力として所定の計算式により値ＭＫを求めるマスター鍵生成手段と

を備え、

前記サーバは、

サーバ登録用のパスワード認証データＨを予め記憶しておく記憶手段と、

前記記憶手段から読み出したパスワード認証データＨと内部において発生させた乱数を入力として所定の計算式により値Ｙ２を求め、前記端末装置へ送信するマスク演算手段と、

前記パスワード認証データＨと内部において発生させた乱数と前記端末装置から受信した値Ｙ１を入力として所定の計算式により値ＭＫを求めるマスター鍵生成手段と

を備えたことを特徴とする認証システム。

2． ユーザが予め決定したパスワードに基づいて、前記パスワード認証データＨと前記認証情報Ｐ’を求めるデータ伸長手段をさらに備えたことを特徴とする請求項１に記載の認証システム。

3． 前記端末装置は、

前記値ＭＫを入力として所定の計算式により値Ｖ１を求め、サーバへ送信するとともに、前記サーバから受信した値Ｖ２と前記値ＭＫを入力として所定の計算式による値Ｖ２と比較照合し、一致した場合にサーバを認証する認証結果

判断手段をさらに備え、

前記サーバは、

前記値ＭＫを入力として所定の計算式により値Ｖ２を求め、端末装置へ送信するとともに、前記端末装置から受信した値Ｖ１と前記値ＭＫを入力として所定の計算式による値Ｖ１と比較照合し、一致した場合に端末装置を認証する認証結果判断手段をさらに備えた

ことを特徴とする請求項１または２に記載の認証システム。

４．　前記端末装置及びサーバは、相互の認証が行われた場合に、セッション鍵を生成するセッション鍵生成手段をそれぞれに備えたことを特徴とする請求項３に記載の認証システム。

５．　前記認証情報Ｐ’は、多項式であることを特徴とする請求項１から４のいずれかに記載の認証システム。

６．　前記認証情報Ｐ’は、多項式とハッシュ関数であることを特徴とする請求項１から４のいずれかに記載の認証システム。

７．　前記認証情報Ｐ’は、ハッシュ関数であることを特徴とする請求項１から４のいずれかに記載の認証システム。

８．　前記認証情報Ｐ’は、疑似乱数関数であることを特徴とする請求項１から４のいずれかに記載の認証システム。

９．　端末装置とサーバ間において相互に認証を行う認証システムにおける端末装置上で動作する認証プログラムであって、

ユーザ保存用の認証情報Ｐ’を予め記憶しておく記憶処理と、

前記記憶しておいた認証情報Ｐ’と認証時に入力されたパスワードを入力として所定の計算式により値Ｐを求める結合処理と、

前記値Ｐと内部において発生させた乱数を入力として所定の計算式により値Ｙ１を求め、前記サーバへ送信するマスク演算処理と、

前記値Ｐと内部において発生させた乱数と前記サーバから受信した値Ｙ２を入力として所定の計算式により値ＭＫを求めるマスター鍵生成処理と

をコンピュータに行わせることを特徴とする認証プログラム。

１０．　ユーザが予め決定したパスワードに基づいて、前記認証情報Ｐ’を求めるデータ伸長処理をさらにコンピュータに行わせることを特徴とする請求項９に記載の認証プログラム。

１１．　前記値ＭＫを入力として所定の計算式により値Ｖ１を求め、サーバへ送信するとともに、前記サーバから受信した値Ｖ２と前記値ＭＫを入力として所定の計算式による値Ｖ２と比較照合し、一致した場合にサーバを認証する認証結果判断処理をさらにコンピュータに行わせることを特徴とする請求項９または１０に記載の認証プログラム。

１２．　端末装置とサーバ間において相互に認証を行う認証システムにおけるサーバ上で動作する認証プログラムであって、

サーバ登録用のパスワード認証データＨを予め記憶しておく記憶処理と、

前記記憶しておいたパスワード認証データＨと内部において発生させた乱数を入力として所定の計算式により値Ｙ２を求め、前記端末装置へ送信するマスク演算処理と、

前記パスワード認証データＨと内部において発生させた乱数と前記端末装置から受信した値Ｙ１を入力として所定の計算式により値ＭＫを求めるマスター鍵生成処理と

をコンピュータに行わせることを特徴とする認証プログラム。

１３．　ユーザが予め決定したパスワードに基づいて、前記パスワード認証データＨを求めるデータ伸長処理をさらにコンピュータに行わせることを特徴

とする請求項１２に記載の認証プログラム。

１４．　前記値ＭＫを入力として所定の計算式により値Ｖ２を求め、端末装置へ送信するとともに、前記端末装置から受信した値Ｖ１と前記値ＭＫを入力として所定の計算式による値Ｖ１と比較照合し、一致した場合に端末装置を認証する認証結果判断処理をさらにコンピュータに行わせることを特徴とする請求項１２または１３に記載の認証プログラム。

１５．　端末装置及びサーバにおいて相互の認証が行われた場合に、セッション鍵を生成するセッション鍵生成処理をそれぞれに備えたことを特徴とする請求項１１または１４に記載の認証プログラム。

１６．　前記認証情報Ｐ’は、多項式であることを特徴とする請求項９から１５のいずれかに記載の認証プログラム。

１７．　前記認証情報Ｐ’は、多項式とハッシュ関数であることを特徴とする請求項９から１５のいずれかに記載の認証プログラム。

１８．　前記認証情報Ｐ’は、ハッシュ関数であることを特徴とする請求項９から１５のいずれかに記載の認証プログラム。

１９．　前記認証情報Ｐ’は、疑似乱数関数であることを特徴とする請求項９から１５のいずれかに記載の認証プログラム。

２０．　前記端末装置は、

更新情報Ｔ’を発生させる発生手段と、

前記記憶手段に記憶されている認証情報Ｐ’と前記更新情報Ｔ’を入力し、所定の計算式によってサーバ更新用のパスワード認証データＨ’と新たな認証情報Ｐ’を求め、サーバ更新用のパスワード認証データＨ’を前記サーバへ送

信するとともに、新たな認証情報Ｐ’を前記記憶手段に記憶する更新情報生成手段と

を備え、

前記サーバは、

前記端末装置から送信されたサーバ更新用のパスワード認証データＨ’と前記記憶手段に記憶されたパスワード認証データＨを入力し、所定の計算式によって新たなパスワード認証データＨを求めて前記記憶手段に記憶されているパスワード認証データＨを更新する更新情報生成手段を備えた

ことを特徴とする請求項２に記載の認証システム。

２１．　前記端末装置は、

秘密情報Ｓ’を発生させる発生手段と、

前記記憶手段に記憶されている認証情報Ｐ’と前記秘密情報Ｓ’と新しいパスワードを入力し、所定の計算式によってサーバ更新用のパスワード認証データＨ’と新たな認証情報Ｐ’を求め、サーバ更新用のパスワード認証データＨ’を前記サーバへ送信するとともに、新たな認証情報Ｐ’を前記記憶手段に記憶する更新情報生成手段と

を備え、

前記サーバは、

前記端末装置から送信されたサーバ更新用のパスワード認証データＨ’と前記記憶手段に記憶されたパスワード認証データＨを入力し、所定の計算式によって新たなパスワード認証データＨを求めて前記記憶手段に記憶されているパスワード認証データＨを更新する更新情報生成手段を備えた

ことを特徴とする請求項２に記載の認証システム。

２２．　端末装置とサーバ間において相互に認証を行う認証システムであって、

前記端末装置は、

ユーザ保存用の認証情報Ｐ’とＲＳＡ公開鍵（Ｎ，ｅ）を予め記憶しておく記憶手段と、

前記記憶手段から読み出した認証情報Ｐ’と認証時に入力されたパスワードを入力として所定の計算式により値Ｗを求める結合手段と、

前記値Ｗと前記記憶手段から読み出したＲＳＡ公開鍵（Ｎ，ｅ）と内部において発生させた乱数Ｔを入力として所定の計算式により値Ｚを求め、前記サーバへ送信するマスク演算手段と

を備え、

前記サーバは、

サーバ登録用のパスワード認証データＨとＲＳＡ秘密鍵（Ｎ，ｄ）を予め記憶しておく記憶手段と、

前記記憶手段から読み出したパスワード認証データＨとＲＳＡ秘密鍵（Ｎ，ｄ）と前記端末装置から受信した値Ｚを入力として所定の計算式により値Ｔを求めるマスター鍵生成手段と

を備えたことを特徴とする認証システム。

２３．　ユーザが予め決定したパスワードに基づいて、前記パスワード認証データＨと前記認証情報Ｐ’を求めるデータ伸長手段をさらに備えたことを特徴とする請求項２２に記載の認証システム。

２４．　前記ＲＳＡ公開鍵（Ｎ，ｅ）と前記ＲＳＡ秘密鍵（Ｎ，ｄ）を求めるＲＳＡ鍵生成手段をさらに備えたことを特徴とする請求項２２に記載の認証システム。

２５．　前記端末装置は、

前記サーバから受信した値Ｖ２と前記乱数Ｔを入力として所定の計算式による値Ｖ２と比較照合し、一致した場合にサーバを認証する認証結果判断手段と、

前記乱数Ｔを入力として所定の計算式により値Ｖ１を求め、前記サーバへ送信する検証子生成手段をさらに備え、

前記サーバは、

前記値Ｔを入力として所定の計算式により値Ｖ２を求め、前記端末装置へ送信する検証子生成手段と、

前記端末装置から受信した値Ｖ１と前記値Ｔを入力として所定の計算式による値Ｖ１と比較照合し、一致した場合に端末装置を認証する認証結果判断手段をさらに備えた

ことを特徴とする請求項２２、２３または２４に記載の認証システム。

２６．　前記端末装置及びサーバは、相互の認証が行われた場合に、セッション鍵を生成するセッション鍵生成手段をそれぞれに備えたことを特徴とする請求項２５に記載の認証システム。

２７．　前記認証情報Ｐ’は、多項式とＦＤＨ関数であることを特徴とする請求項２２から２６のいずれかに記載の認証システム。

２８．　前記認証情報Ｐ’は、ＦＤＨ関数であることを特徴とする請求項２２から２６のいずれかに記載の認証システム。

２９．　前記ＲＳＡ公開鍵（Ｎ，ｅ）は、安全な通信を用いることを特徴とする請求項２２から２６のいずれかに記載の認証システム。

３０．　前記ＲＳＡ公開鍵（Ｎ，ｅ）は、安全ではない通信を用いることを特徴とする請求項２２から２６のいずれかに記載の認証システム。

３１．　端末装置とサーバ間において相互に認証を行う認証システムにおける端末装置上で動作する認証プログラムであって、

ユーザ保存用の認証情報Ｐ’とＲＳＡ公開鍵（Ｎ，ｅ）を予め記憶しておく記憶処理と、

前記記憶しておいた認証情報Ｐ’と認証時に入力されたパスワードを入力として所定の計算式により値Ｗを求める結合処理と、

前記値Wと前記記憶しておいたＲＳＡ公開鍵（Ｎ，ｅ）と内部において発生させた乱数Ｔを入力として所定の計算式により値Ｚを求め、前記サーバへ送信するマスク演算処理と

をコンピュータに行わせることを特徴とする認証プログラム。

３２．　ユーザが予め決定したパスワードに基づいて、前記認証情報Ｐ’を求めるデータ伸長処理をさらにコンピュータに行わせることを特徴とする請求項３１に記載の認証プログラム。

３３．　前記ＲＳＡ公開鍵（Ｎ，ｅ）を求めるＲＳＡ鍵生成処理をさらにコンピュータに行わせることを特徴とする請求項３１に記載の認証プログラム。

３４．　前記サーバから受信した値Ｖ２と前記乱数Ｔを入力として所定の計算式による値Ｖ２と比較照合し、一致した場合にサーバを認証する認証結果判断処理と、

前記乱数Ｔを入力として所定の計算式により値Ｖ１を求め、前記サーバへ送信する検証子生成処理をさらにコンピュータに行わせることを特徴とする請求項３１、３２または３３に記載の認証プログラム。

３５．　端末装置とサーバ間において相互に認証を行う認証システムにおけるサーバ上で動作する認証プログラムであって、

サーバ登録用のパスワード認証データＨとＲＳＡ秘密鍵（Ｎ，ｄ）を予め記憶しておく記憶処理と、

前記記憶しておいたパスワード認証データＨとＲＳＡ秘密鍵（Ｎ，ｄ）と前記端末装置から受信した値Ｚを入力として所定の計算式により値Ｔを求めるマスター鍵生成処理と

をコンピュータに行わせることを特徴とする認証プログラム。

３６．　ユーザが予め決定したパスワードに基づいて、前記パスワード認証デ

ータＨを求めるデータ伸長処理をさらにコンピュータに行わせることを特徴とする請求項３５に記載の認証プログラム。

３７．　前記ＲＳＡ秘密鍵（Ｎ，ｄ）を求めるＲＳＡ鍵生成処理をさらにコンピュータに行わせることを特徴とする請求項３５に記載の認証プログラム。

３８．　前記値Ｔを入力として所定の計算式により値Ｖ２を求め、前記端末装置へ送信する検証子生成処理と、

前記端末装置から受信した値Ｖ１と前記値Ｔを入力として所定の計算式による値Ｖ１と比較照合し、一致した場合に端末装置を認証する認証結果判断処理をさらにコンピュータに行わせることを特徴とする請求項３５、３６または３７に記載の認証プログラム。

３９．　端末装置及びサーバにおいて相互の認証が行われた場合に、セッション鍵を生成するセッション鍵生成処理をそれぞれに備えたことを特徴とする請求項３４または３８に記載の認証プログラム。

４０．　前記認証情報Ｐ’は、多項式とＦＤＨ関数であることを特徴とする請求項３１から３９のいずれかに記載の認証プログラム。

４１．　前記認証情報Ｐ’は、ＦＤＨ関数であることを特徴とする請求項３１から３９のいずれかに記載の認証プログラム。

４２．　前記ＲＳＡ公開鍵（Ｎ，ｅ）は、安全な通信を用いることを特徴とする請求項３１から３９のいずれかに記載の認証プログラム。

４３．　前記ＲＳＡ公開鍵（Ｎ，ｅ）は、安全ではない通信を用いることを特徴とする請求項３１から３９のいずれかに記載の認証プログラム。

４４．　前記端末装置は、

更新情報Ｔ’を発生させる発生手段と、

前記記憶手段に記憶されている認証情報Ｐ’と前記更新情報Ｔ’を入力し、所定の計算式によってサーバ更新用のパスワード認証データＨ’と新たな認証情報Ｐ’を求め、サーバ更新用のパスワード認証データＨ’を前記サーバへ送信するとともに、新たな認証情報Ｐ’を前記記憶手段に記憶する更新情報生成手段と

を備え、

前記サーバは、

前記端末装置から送信されたサーバ更新用のパスワード認証データＨ’と前記記憶手段に記憶されたパスワード認証データＨを入力し、所定の計算式によって新たなパスワード認証データＨを求めて前記記憶手段に記憶されているパスワード認証データＨを更新する更新情報生成手段を備えた

ことを特徴とする請求項２３に記載の認証システム。

４５．　前記端末装置は、

前記記憶手段に記憶されている認証情報Ｐ’と前記乱数Ｔを入力し、所定の計算式によって新たな認証情報Ｐ’を求め、新たな認証情報Ｐ’を前記記憶手段に記憶する更新情報生成手段と

を備え、

前記サーバは、

前記記憶手段に記憶されたパスワード認証データＨと前記マスター鍵生成手段によって求めた値Ｔを入力し、所定の計算式によって新たなパスワード認証データＨを求めて前記記憶手段に記憶されているパスワード認証データＨを更新する更新情報生成手段を備えた

ことを特徴とする請求項２２に記載の認証システム。

４６．　前記端末装置は、

秘密情報Ｓ’を発生させる発生手段と、

前記記憶手段に記憶されている認証情報Ｐ’と前記秘密情報Ｓ’と新しいパスワードを入力し、所定の計算式によってサーバ更新用のパスワード認証データＨ’と新たな認証情報Ｐ’を求め、サーバ更新用のパスワード認証データＨ’を前記サーバへ送信するとともに、新たな認証情報Ｐ’を前記記憶手段に記憶する更新情報生成手段と

を備え、

前記サーバは、

前記端末装置から送信されたサーバ更新用のパスワード認証データＨ’と前記記憶手段に記憶されたパスワード認証データＨを入力し、所定の計算式によって新たなパスワード認証データＨを求めて前記記憶手段に記憶されているパスワード認証データＨを更新する更新情報生成手段を備えた

ことを特徴とする請求項２３に記載の認証システム。

４７．　端末装置と複数のサーバ間において相互に認証を行い、前記端末装置内の保存対象のデータを前記サーバ内に分散して保存する遠隔分散保存システムであって、

前記端末装置は、

ユーザが予め決定しておいたパスワードに基づいて、サーバ登録用のパスワード認証データＨとユーザ保存用の認証情報Ｐ’を求めるデータ伸長手段と、

前記データ伸長手段によって求めた認証情報Ｐ’を予め記憶しておく記憶手段と、

前記記憶手段から読み出した認証情報Ｐ’と認証時に入力されたパスワードを入力として所定の計算式により値Ｐを求める結合手段と、

前記値Ｐと内部において発生させた乱数を入力として所定の計算式により値Ｙ１を求め、前記サーバへ送信するマスク演算手段と、

前記値Ｐと内部において発生させた乱数と前記サーバから受信した値Ｙ２を入力として所定の計算式により値ＭＫを求めるマスター鍵生成手段と、

前記値ＭＫを入力として所定の計算式により値Ｖ１を求め、サーバへ送信するとともに、前記サーバから受信した値Ｖ２と値Ｖ１と比較照合し、一致した場合

にサーバを認証する認証結果判断手段と、

サーバの認証が行われた場合に、セッション鍵ＳＫをサーバの数だけ生成するセッション鍵生成手段と、

前記保存対象のデータを分割して、認証したサーバの数と同数の分割データを得るデータ分割手段と、

前記分割データのそれぞれと保存対象のデータを識別する識別情報とを、保存先のサーバと共有した前記セッション鍵ＳＫを用いて暗号化して、各サーバに対して送信するデータ保存手段と、

分割データが保存された各サーバから分割データを受信し、前記保存対象のデータを復元するデータ復元手段と

を備え、

前記サーバは、

前記データ伸長手段によって求めたパスワード認証データＨを予め記憶しておく記憶手段と、

前記記憶手段から読み出したパスワード認証データＨと内部において発生させた乱数を入力として所定の計算式により値Ｙ２を求め、前記端末装置へ送信するマスク演算手段と、

前記パスワード認証データＨと内部において発生させた乱数と前記端末装置から受信した値Ｙ１を入力として所定の計算式により値ＭＫを求めるマスター鍵生成手段と、

前記値ＭＫを入力として所定の計算式により値Ｖ２を求め、端末装置へ送信するとともに、前記端末装置から受信した値Ｖ１と値Ｖ２と比較照合し、一致した場合に端末装置を認証する認証結果判断手段と、

端末装置の認証が行われた場合に、セッション鍵を生成するセッション鍵生成手段と、

端末装置から受信した分割データを受信するデータ受信手段と、

前記分割データを記憶するデータ記憶手段と、

前記データ記憶手段に保存されている分割データを読み出して端末装置へ送信するデータ送信手段と

を備えたことを特徴とする遠隔分散保存システム。

４８．　前記分割データの一部を前記端末装置内に保存することを特徴とする請求項４７に記載の遠隔分散保存システム。

４９．　端末装置と複数のサーバ間において相互に認証を行い、前記端末装置内の保存対象のデータを前記サーバ内に分散して保存する遠隔分散保存システムにおける端末装置上で動作する遠隔分散保存プログラムであって、

ユーザが予め決定しておいたパスワードに基づいて、サーバ登録用のパスワード認証データＨとユーザ保存用の認証情報Ｐ’を求めるデータ伸長処理と、

前記データ伸長処理によって求めた認証情報Ｐ’を予め記憶しておく記憶処理と、

前記記憶処理から読み出した認証情報Ｐ’と認証時に入力されたパスワードを入力として所定の計算式により値Ｐを求める結合処理と、

前記値Ｐと内部において発生させた乱数を入力として所定の計算式により値Ｙ１を求め、前記サーバへ送信するマスク演算処理と、

前記値Ｐと内部において発生させた乱数と前記サーバから受信した値Ｙ２を入力として所定の計算式により値ＭＫを求めるマスター鍵生成処理と、

前記値ＭＫを入力として所定の計算式により値Ｖ１を求め、サーバへ送信するとともに、前記サーバから受信した値Ｖ２と値Ｖ１と比較照合し、一致した場合にサーバを認証する認証結果判断処理と、

サーバの認証が行われた場合に、セッション鍵ＳＫをサーバの数だけ生成するセッション鍵生成処理と、

前記保存対象のデータを分割して、認証したサーバの数と同数の分割データを得るデータ分割処理と、

前記分割データのそれぞれと保存対象のデータを識別する識別情報とを、保存先のサーバと共有した前記セッション鍵ＳＫを用いて暗号化して、各サーバに対して送信するデータ保存処理と、

分割データが保存された各サーバから分割データを受信し、前記保存対象のデ

ータを復元するデータ復元処理と

をコンピュータに行わせることを特徴とする遠隔分散保存プログラム

５０．　端末装置と複数のサーバ間において相互に認証を行い、前記端末装置内の保存対象のデータを前記サーバ内に分散して保存する遠隔分散保存システムにおけるサーバ上で動作する遠隔分散保存プログラムであって、

データ伸長処理によって求めたパスワード認証データＨを予め記憶しておく記憶処理と、

前記記憶処理から読み出したパスワード認証データＨと内部において発生させた乱数を入力として所定の計算式により値Ｙ２を求め、前記端末装置へ送信するマスク演算処理と、

前記パスワード認証データＨと内部において発生させた乱数と前記端末装置から受信した値Ｙ１を入力として所定の計算式により値ＭＫを求めるマスター鍵生成処理と、

前記値ＭＫを入力として所定の計算式により値Ｖ２を求め、端末装置へ送信するとともに、前記端末装置から受信した値Ｖ１と値Ｖ２と比較照合し、一致した場合に端末装置を認証する認証結果判断処理と、

端末装置の認証が行われた場合に、セッション鍵を生成するセッション鍵生成処理と、

端末装置から受信した分割データを受信するデータ受信処理と、

前記分割データを記憶するデータ記憶処理と、

前記データ記憶処理に保存されている分割データを読み出して端末装置へ送信するデータ送信処理と

をコンピュータに行わせることを特徴とする遠隔分散保存プログラム。

図1

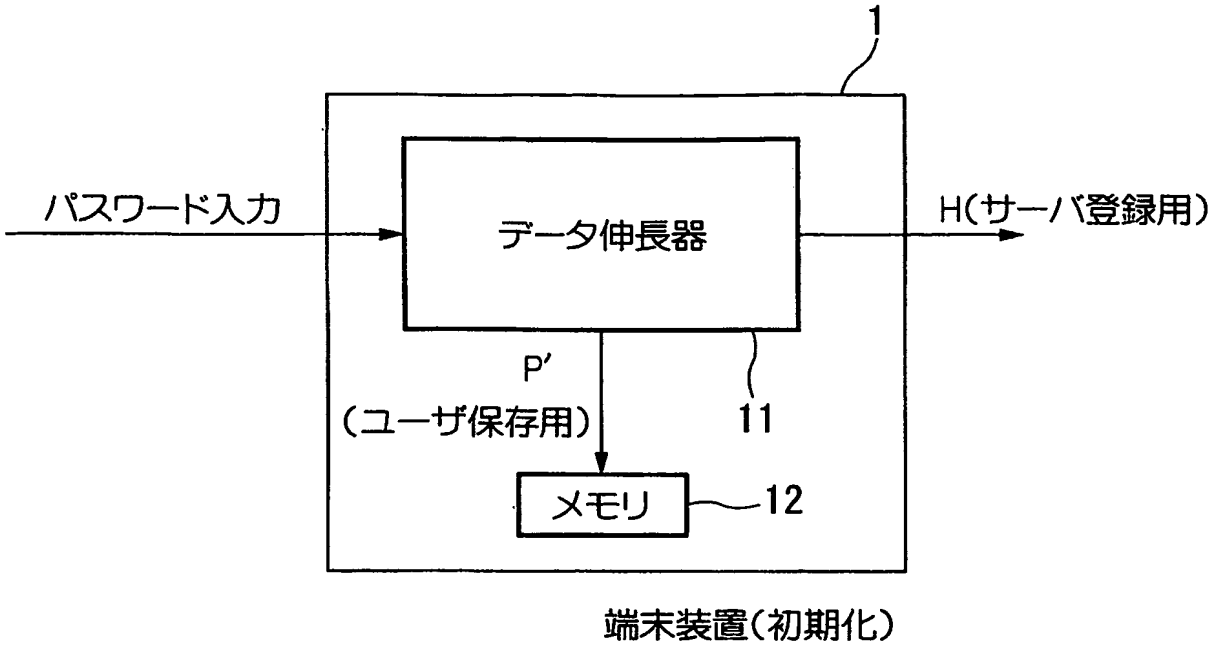

1
パスワード入力
データ伸長器
H(サーバ登録用)
P'
(ユーザ保存用)
11
メモリ
12
端末装置(初期化)

図2

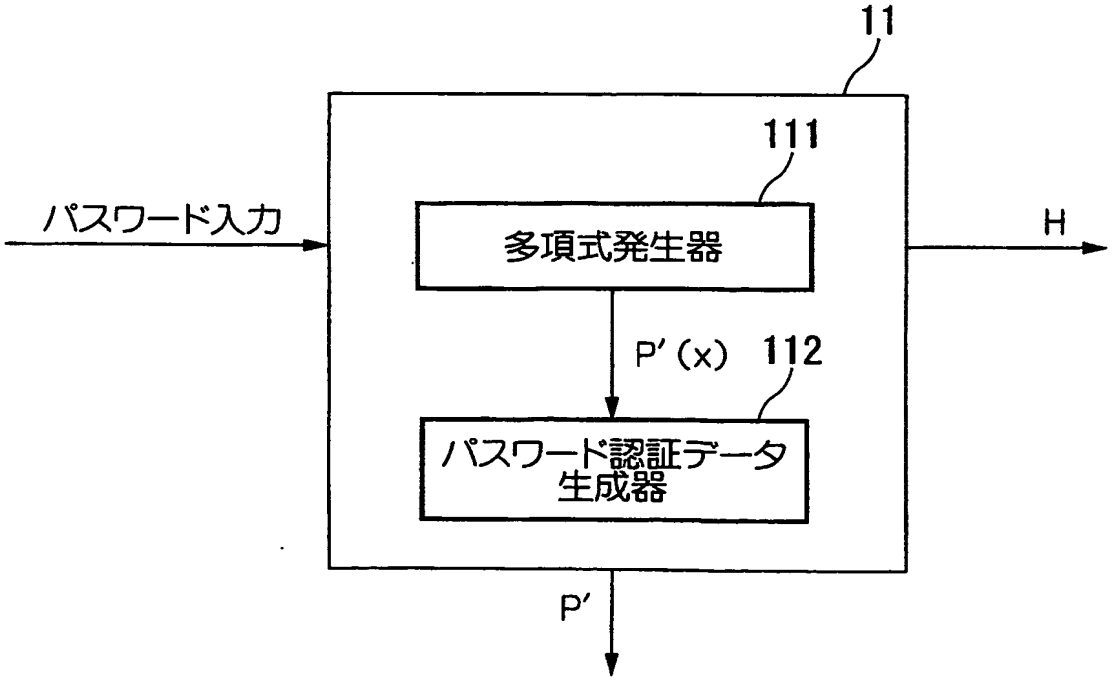

11
111
パスワード入力
多項式発生器
H
P'(x)
112
パスワード認証データ
生成器
P'

図3

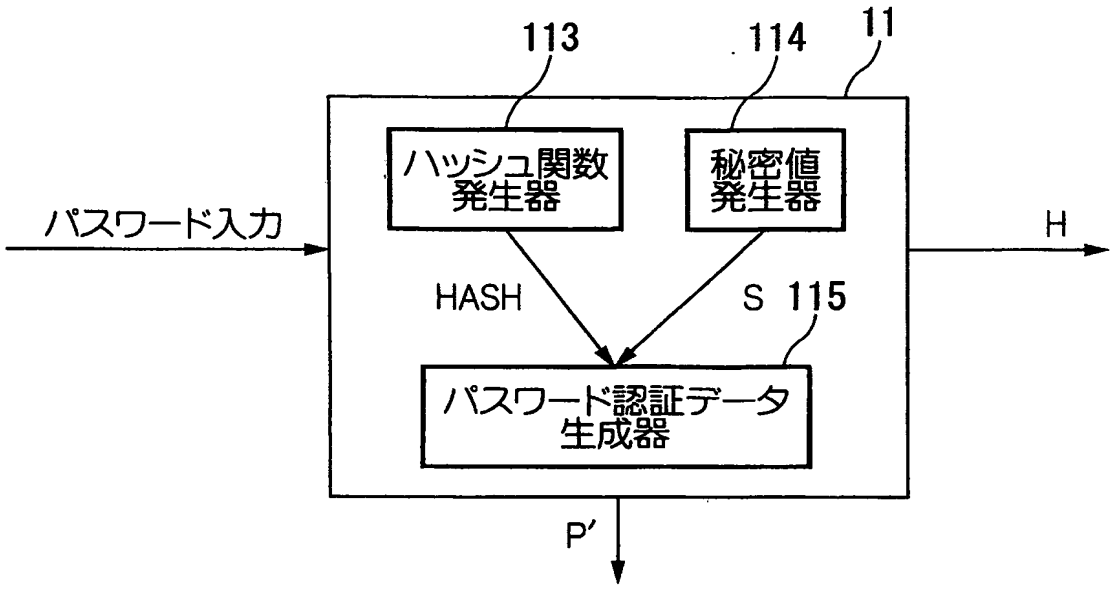

113
114
11
ハッシュ関数
発生器
秘密値
発生器
パスワード入力
H
HASH
S 115
パスワード認証データ
生成器
P'

図4

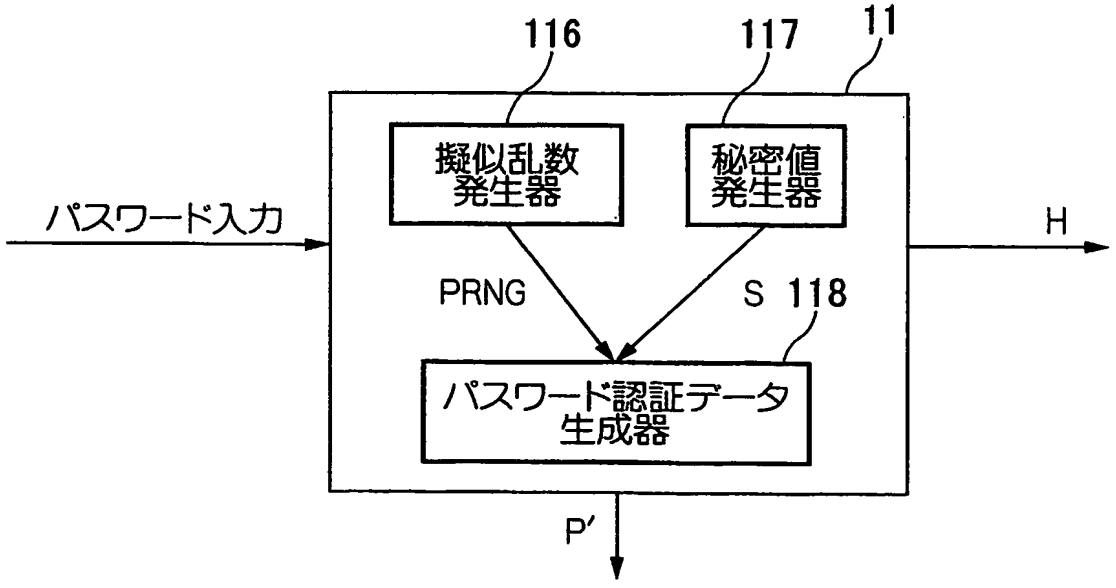
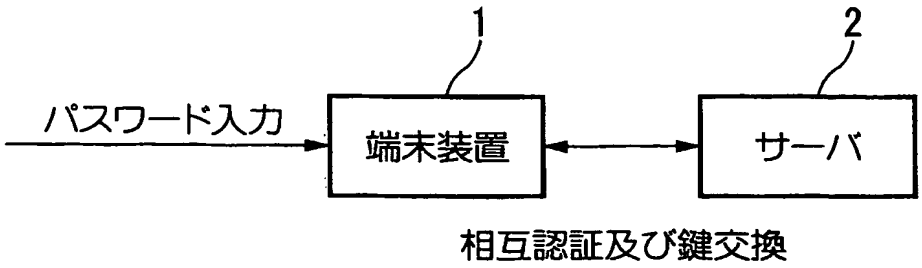

116
117
11
擬似乱数
発生器
秘密値
発生器
パスワード入力
H
PRNG
S 118
パスワード認証データ
生成器
P'

図5



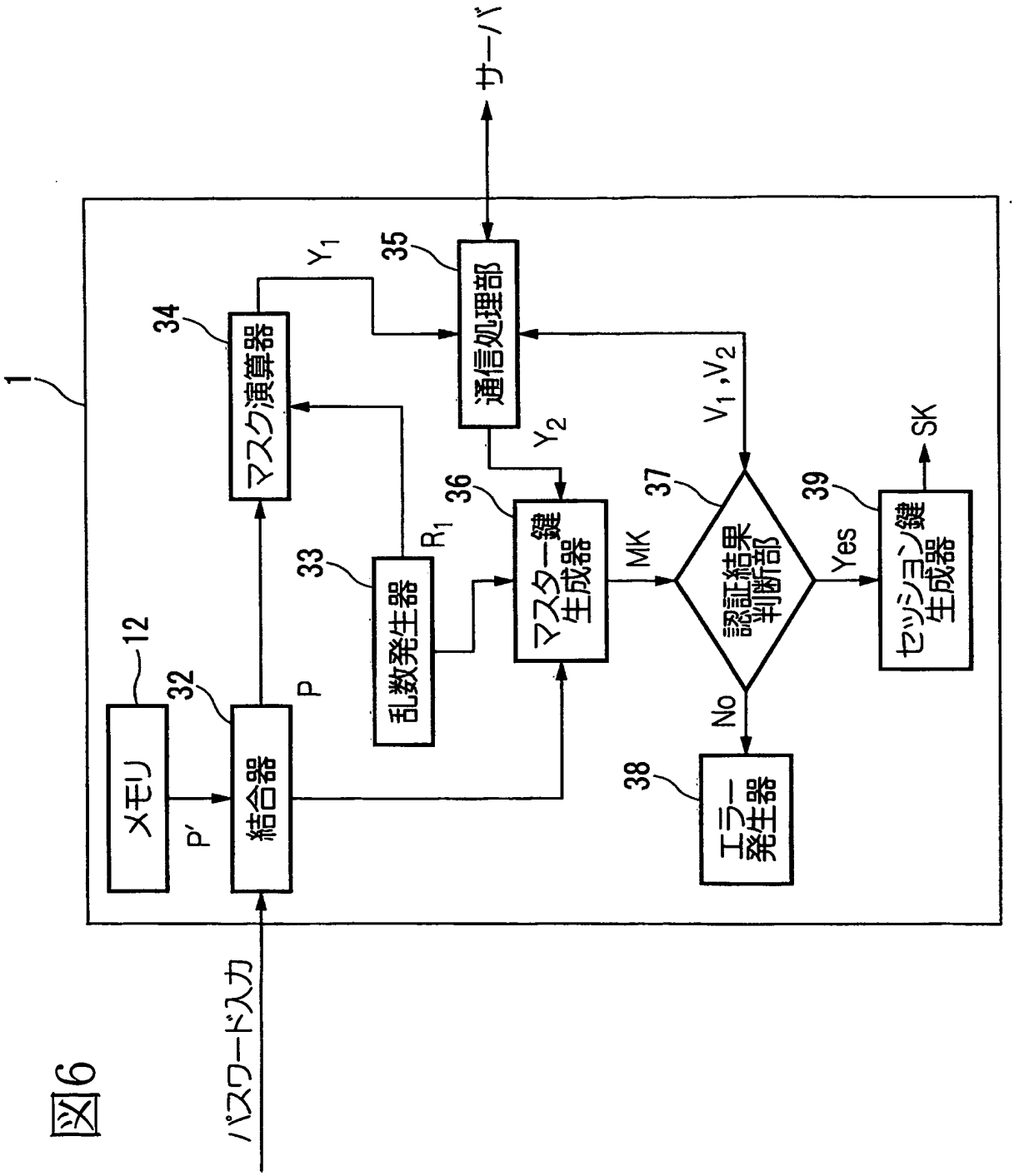

サーバ
1
通信処理部
35
Y_1
Y_2
マスク演算器
34
マスター鍵
生成器
36
乱数発生器
33
R_1
MK
認証結果
判断部
37
V_1, V_2
Yes
No
セッション鍵
生成器
39
SK
エラー
発生器
38
メモリ
12
結合器
32
P
P'
パスワード入力

図6

図7

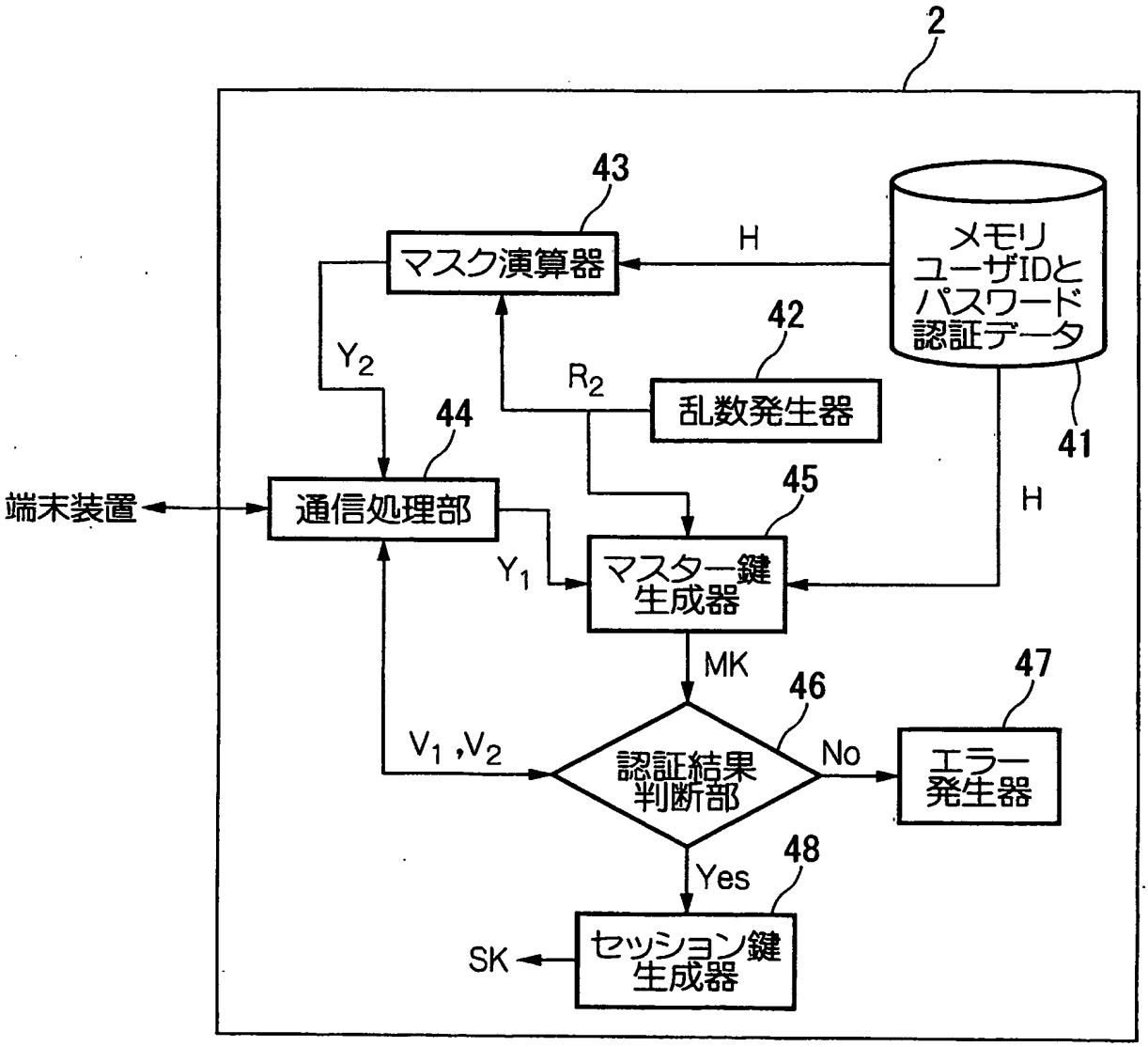

2
43
マスク演算器
H
メモリ
ユーザIDと
パスワード
認証データ
42
乱数発生器
Y_2
R_2
44
41
端末装置
通信処理部
45
H
Y_1
マスター鍵
生成器
MK
47
46
V_1 ,V_2
認証結果
判断部
No
エラー
発生器
48
Yes
SK
セッション鍵
生成器

図8

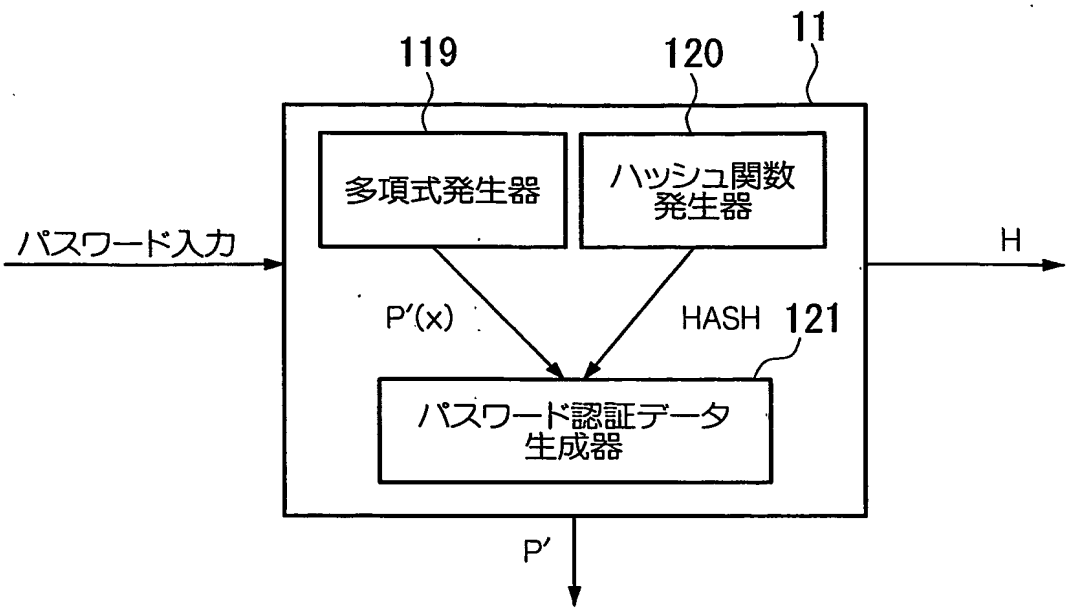

119
120
11
多項式発生器
ハッシュ関数
発生器
パスワード入力
H
P'(x)
HASH
121
パスワード認証データ
生成器
P'

図9

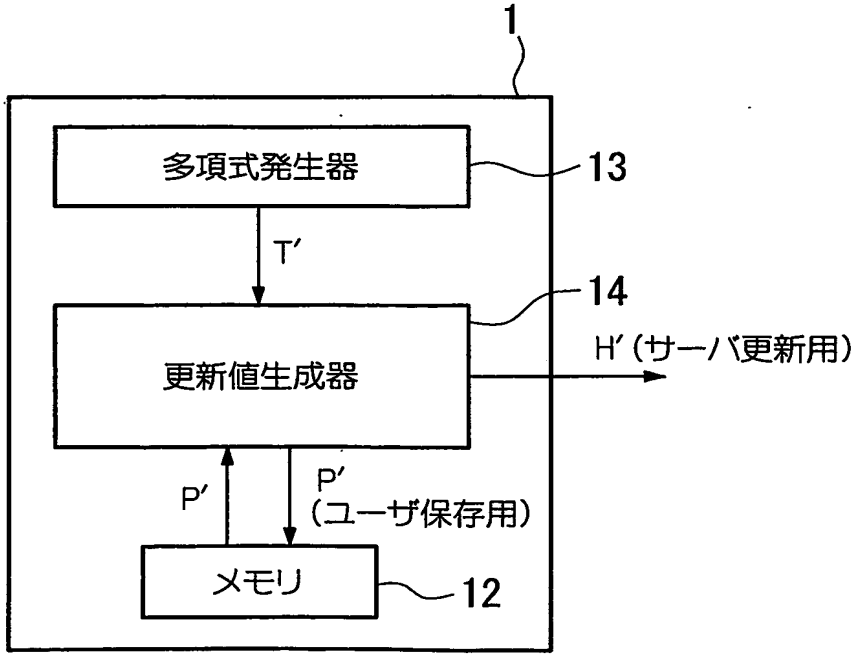

1
多項式発生器
13
T'
14
更新値生成器
H'(サーバ更新用)
P'
P'
(ユーザ保存用)
メモリ
12

図10

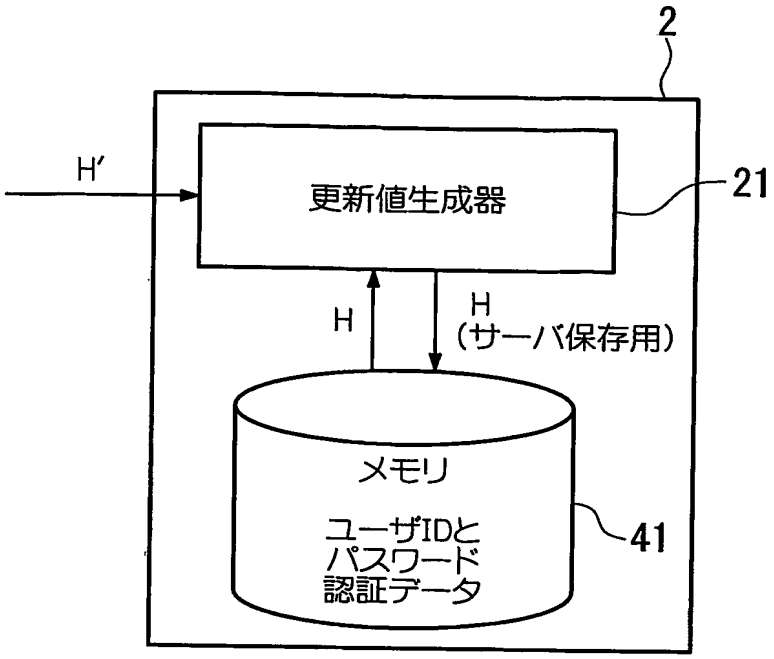

2
H'
更新値生成器
21
H
H
(サーバ保存用)
メモリ
ユーザIDと
パスワード
認証データ
41

図11

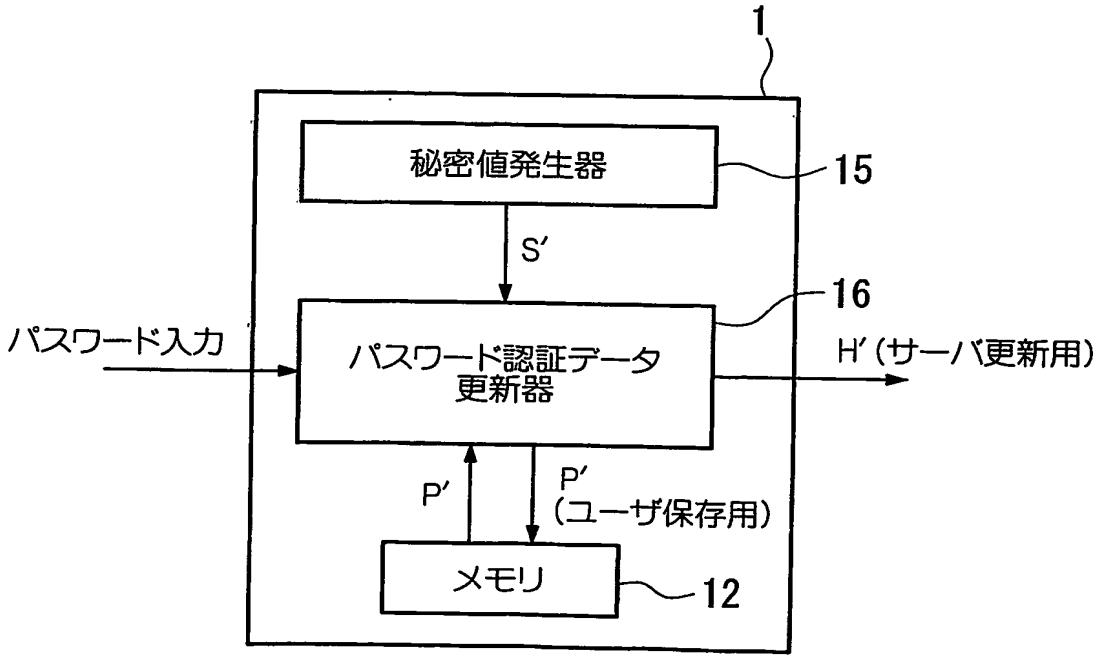

1
秘密値発生器
15
S'
16
パスワード入力
パスワード認証データ
更新器
H'(サーバ更新用)
P'
P'
(ユーザ保存用)
メモリ
12

図12

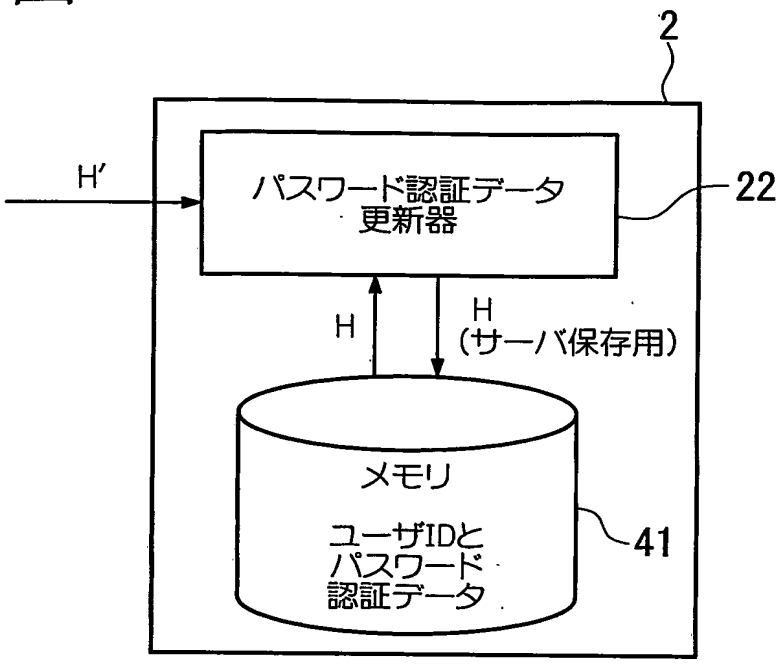

2
H'
パスワード認証データ
更新器
22
H
H
(サーバ保存用)
メモリ
ユーザIDと
パスワード
認証データ
41

図13

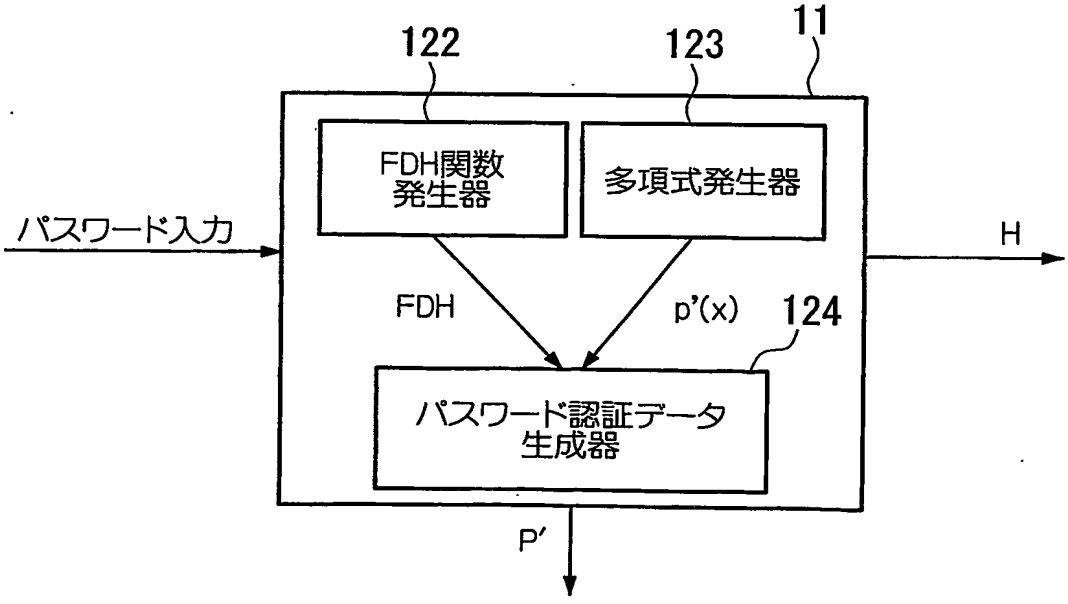

122
123
11
FDH関数
発生器
多項式発生器
パスワード入力
H
FDH
p'(x)
124
パスワード認証データ
生成器
P'

図14

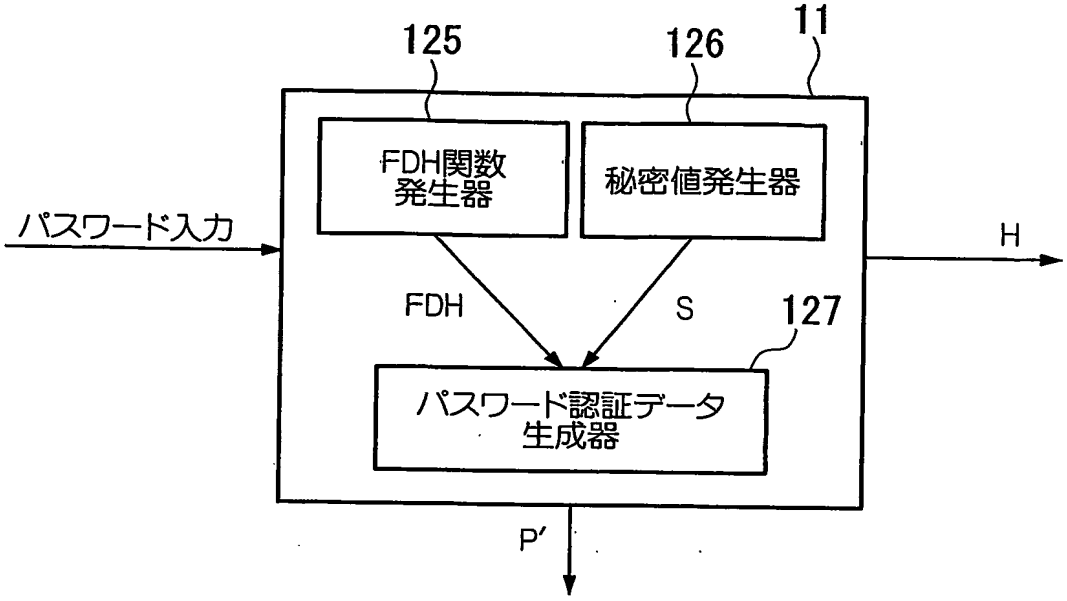
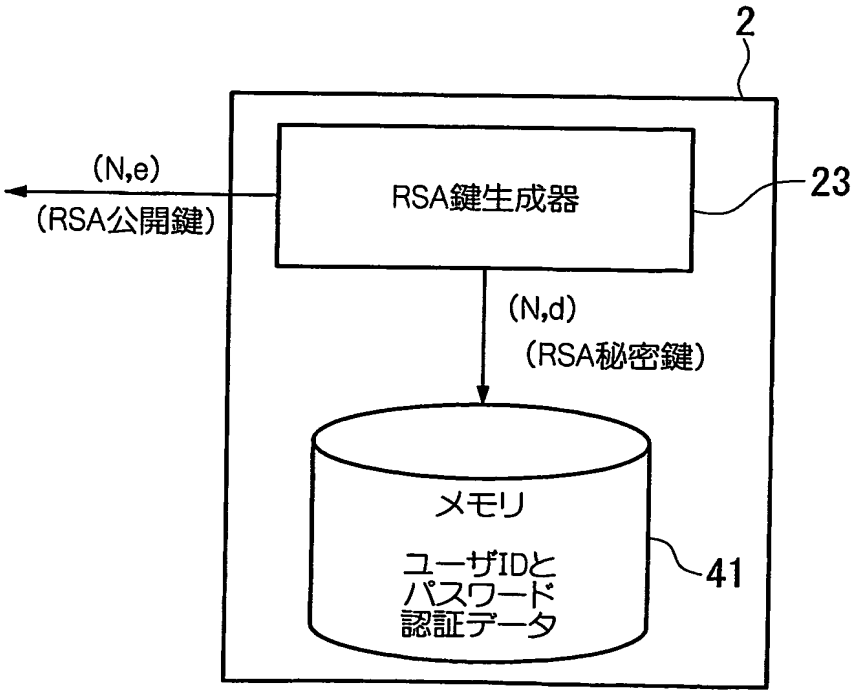

125
126
11
FDH関数
発生器
秘密値発生器
パスワード入力
H
FDH
S
127
パスワード認証データ
生成器
P'

図15


2
(N,e)
(RSA公開鍵)
RSA鍵生成器
23
(N,d)
(RSA秘密鍵)
メモリ
ユーザIDと
パスワード
認証データ
41

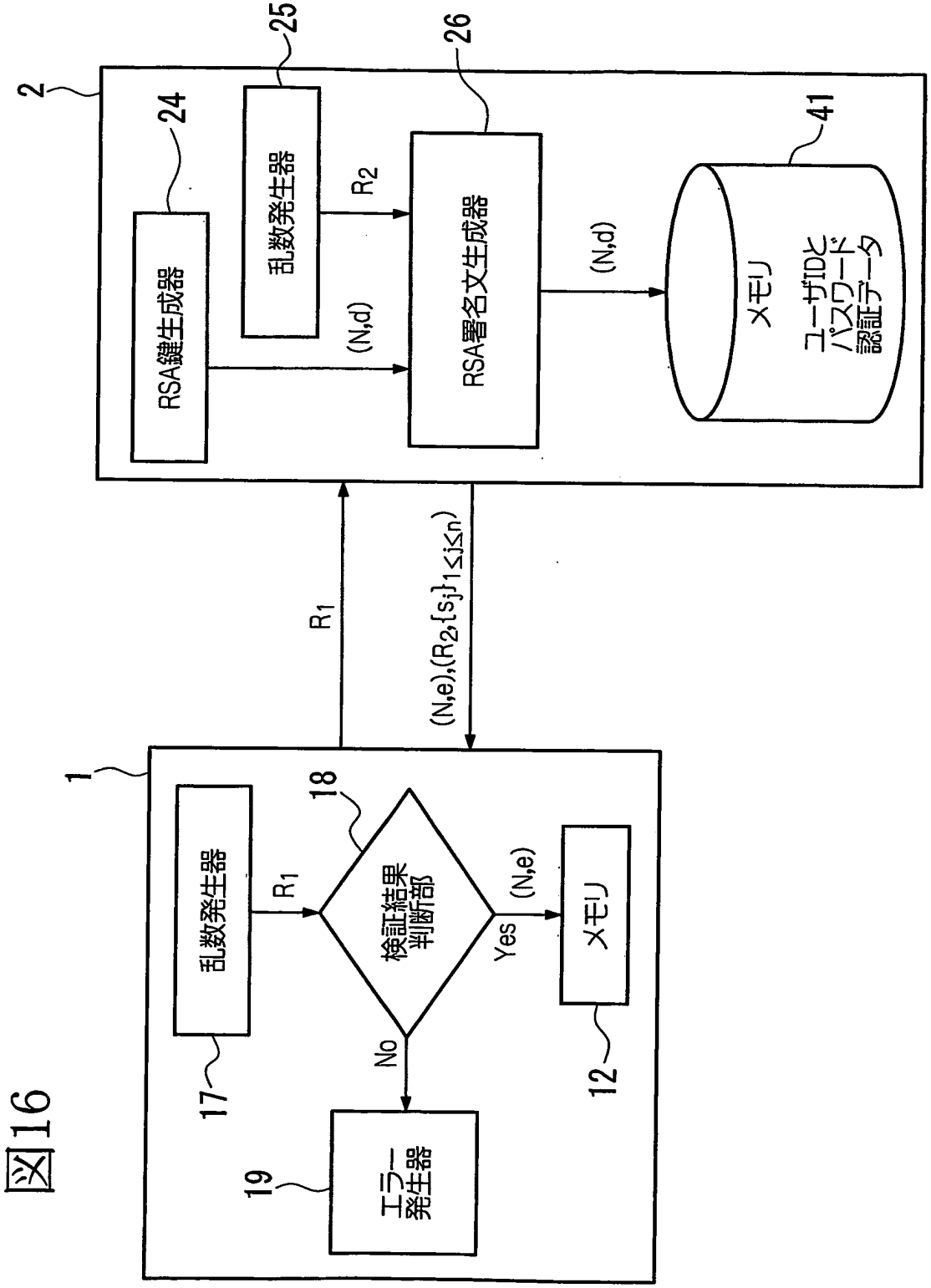

2
25
26
24
41
乱数発生器
RSA鍵生成器
RSA署名文生成器
メモリ
ユーザIDと
パスワード
認証データ
R2
(N,d)
(N,d)
R1
(N,e),(R2,{sj}1≤j≤n)
1
18
乱数発生器
検証結果
判断部
メモリ
R1
(N,e)
Yes
No
17
12
19
エラー
発生器

図16

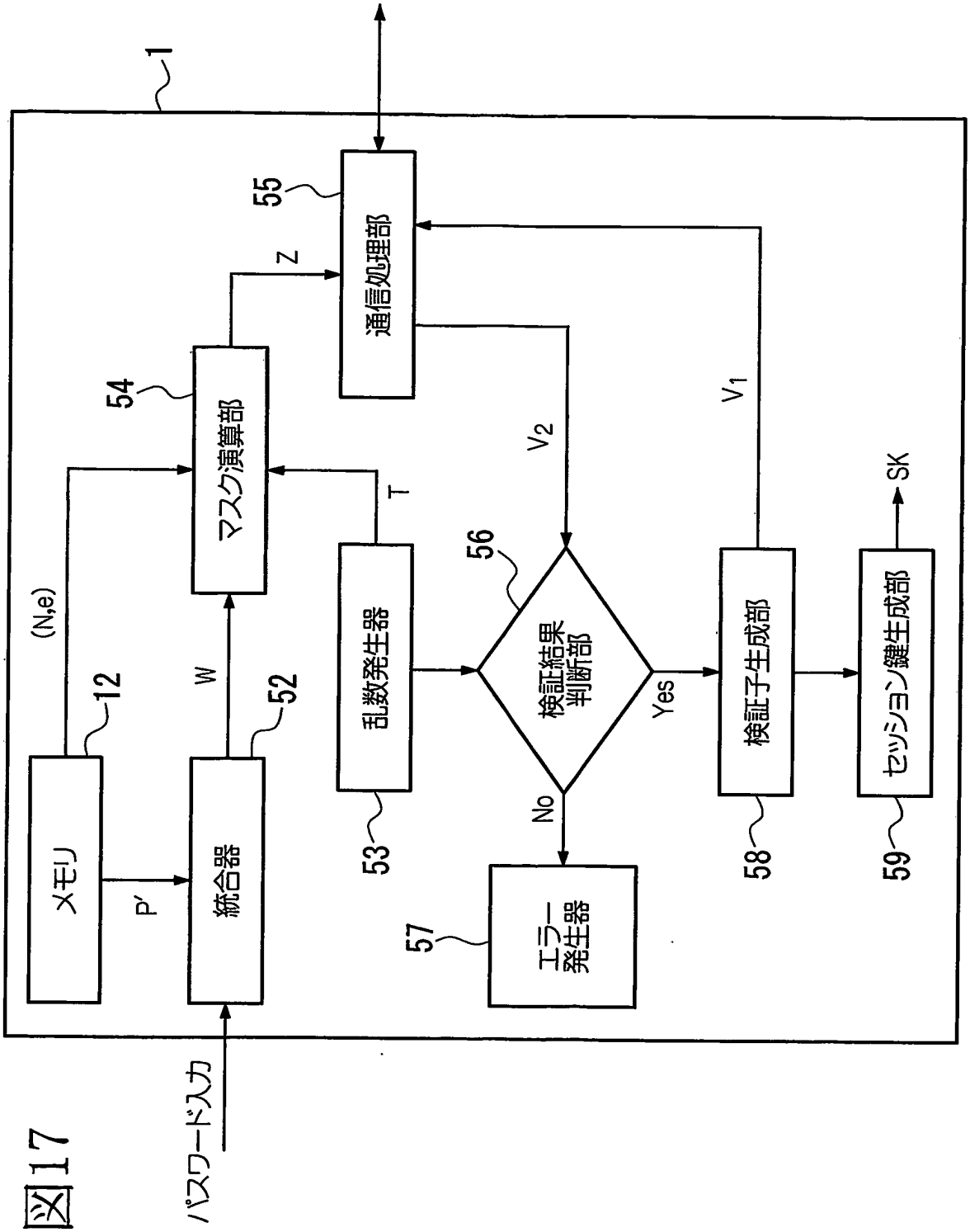

1
55
通信処理部
Z
54
マスク演算部
(N,e)
12
メモリ
P'
W
52
統合器
パスワード入力
T
乱数発生器
53
56
検証結果判断部
No
57
エラー発生器
Yes
V2
V1
検証子生成部
58
SK
セッション鍵生成部
59

図17

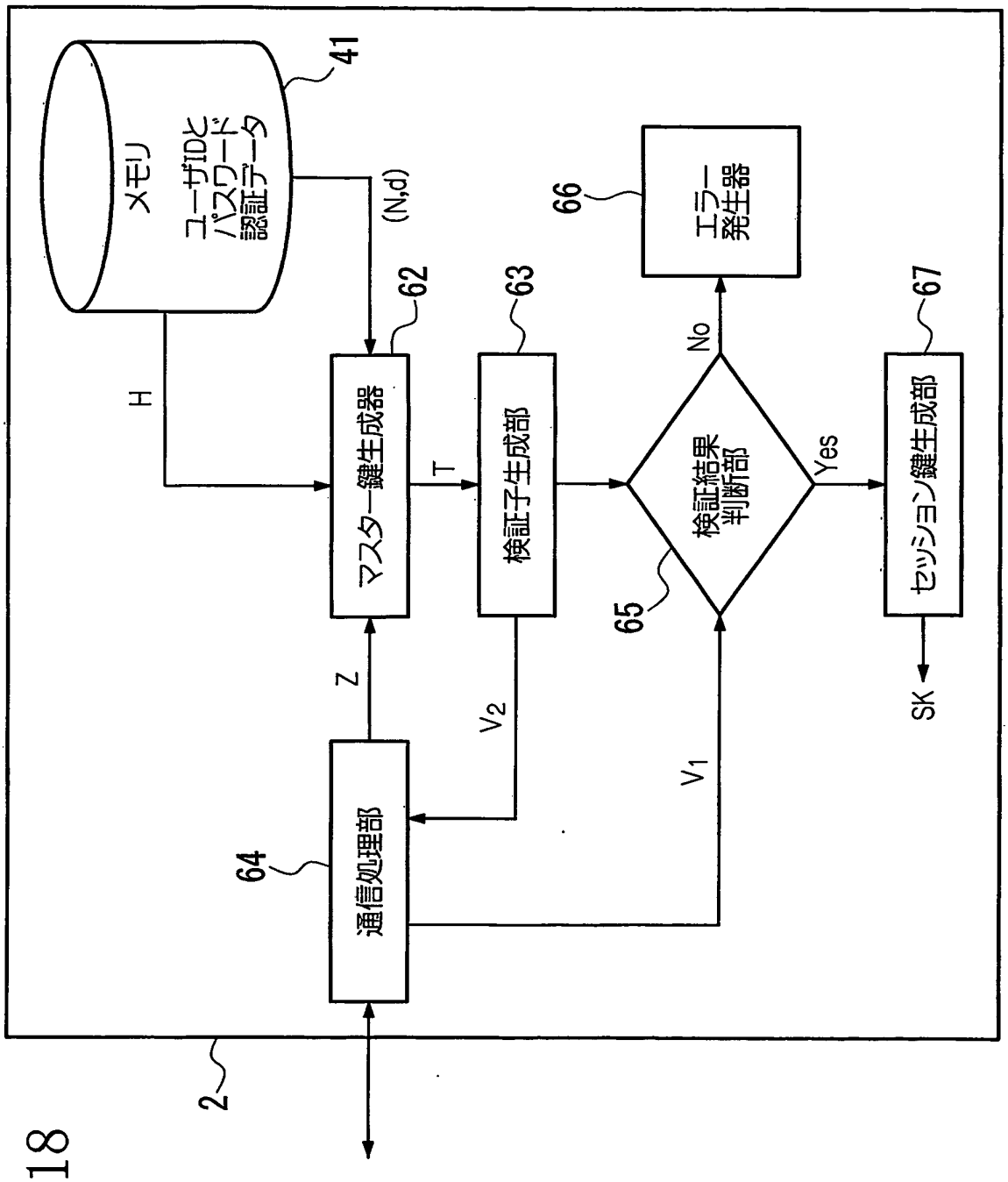

メモリ
ユーザIDと
パスワード
認証データ
41
(N,d)
H
62
マスター鍵生成器
T
63
検証子生成部
66
エラー
発生器
No
検証結果
判断部
65
Yes
67
セッション鍵生成部
SK
Z
V2
V1
64
通信処理部
2

図18

図19

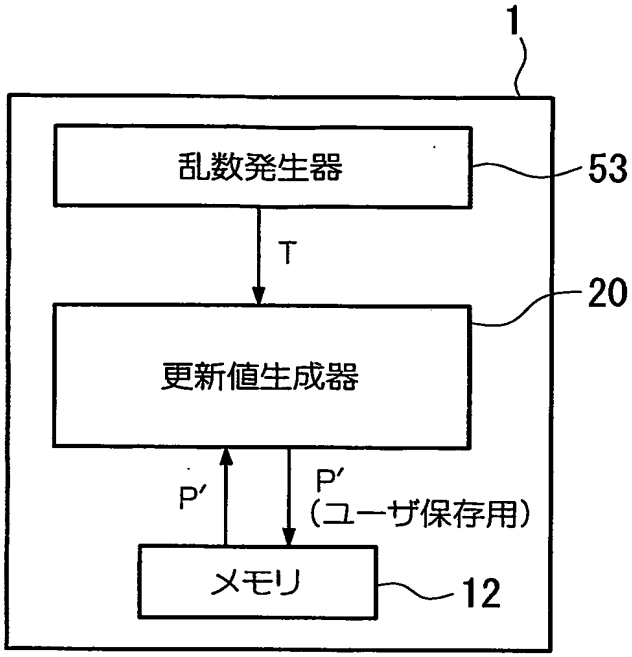


図20

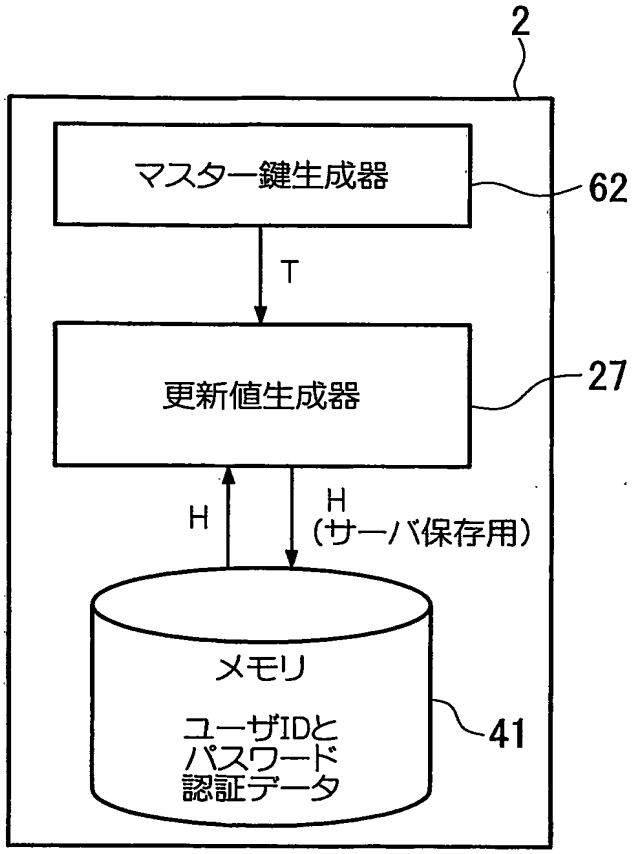


図21

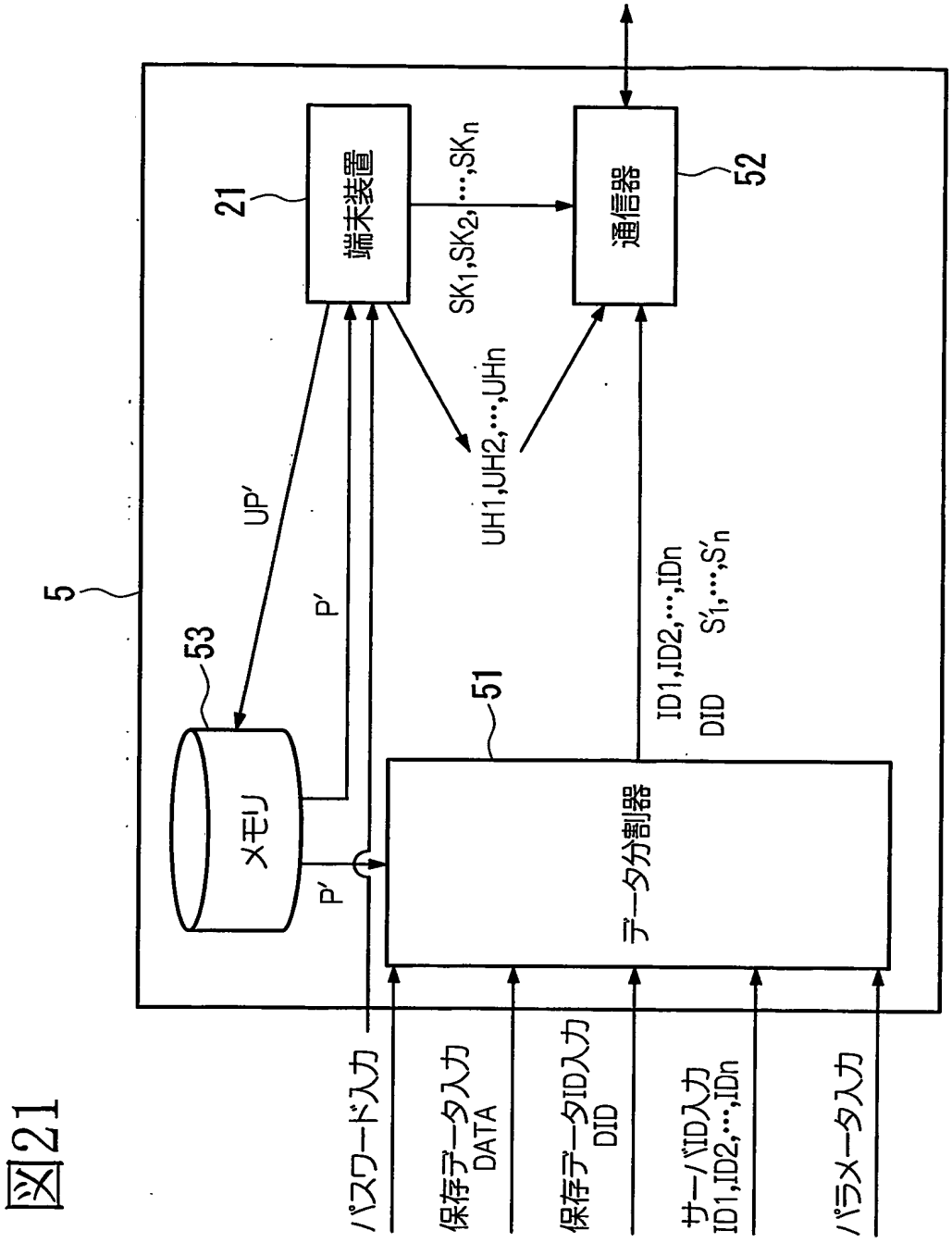
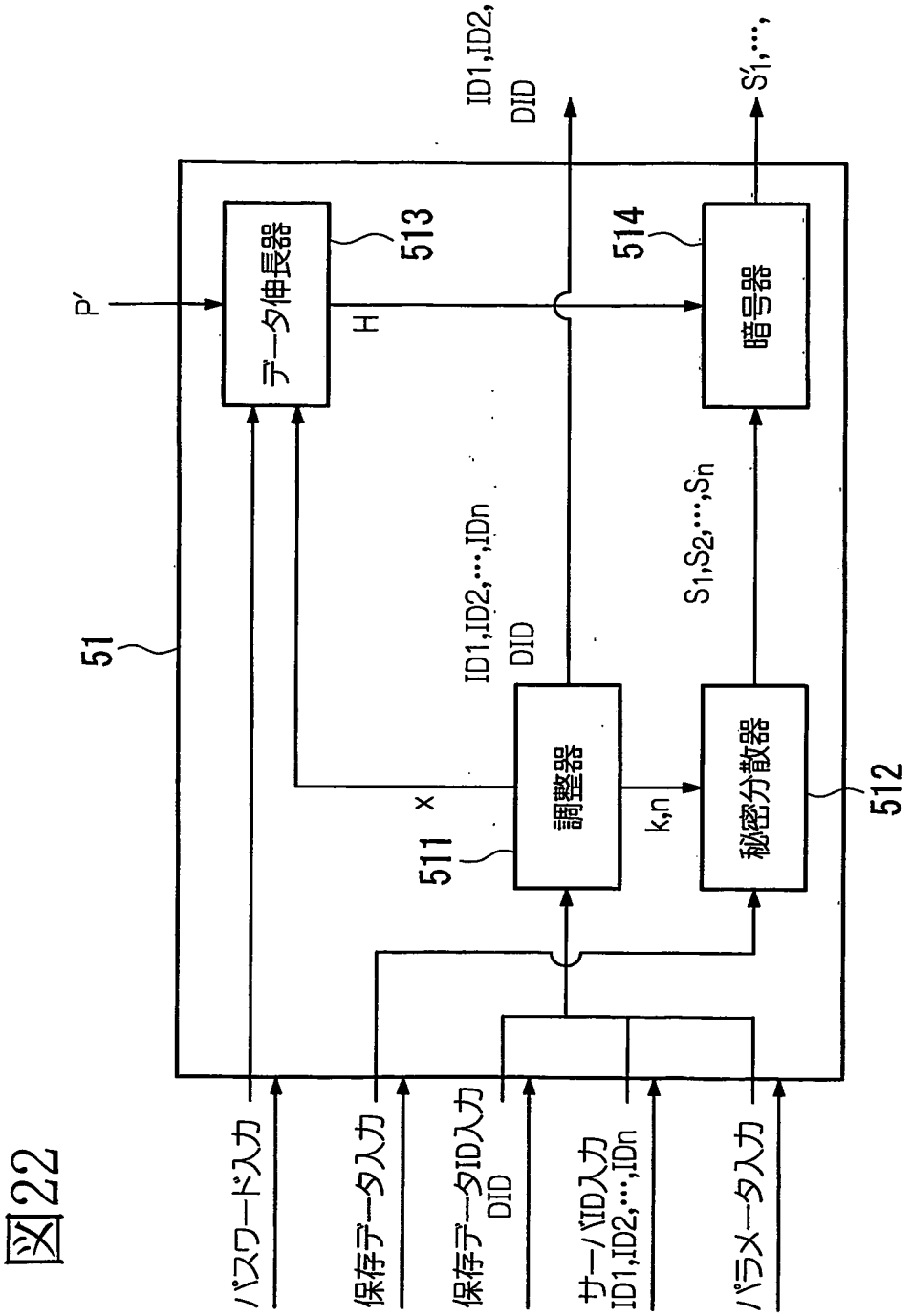

5
端末装置
21
通信器
52
SK1,SK2,…,SKn
UH1,UH2,…,UHn
UP′
P′
53
メモリ
51
データ分割器
ID1,ID2,…,IDn
DID
S′1,…,S′n
パスワード入力
保存データ入力
DATA
保存データID入力
DID
サーバID入力
ID1,ID2,…,IDn
パラメータ入力

図22


51
P'
データ伸長器
513
H
514
暗号器
ID1,ID2,
DID
S'1,…,
ID1,ID2,…,IDn
DID
S1,S2,…,Sn
調整器
秘密分散器
511
x
k,n
512
パスワード入力
保存データ入力
保存データID入力
DID
サーバID入力
ID1,ID2,…,IDn
パラメータ入力

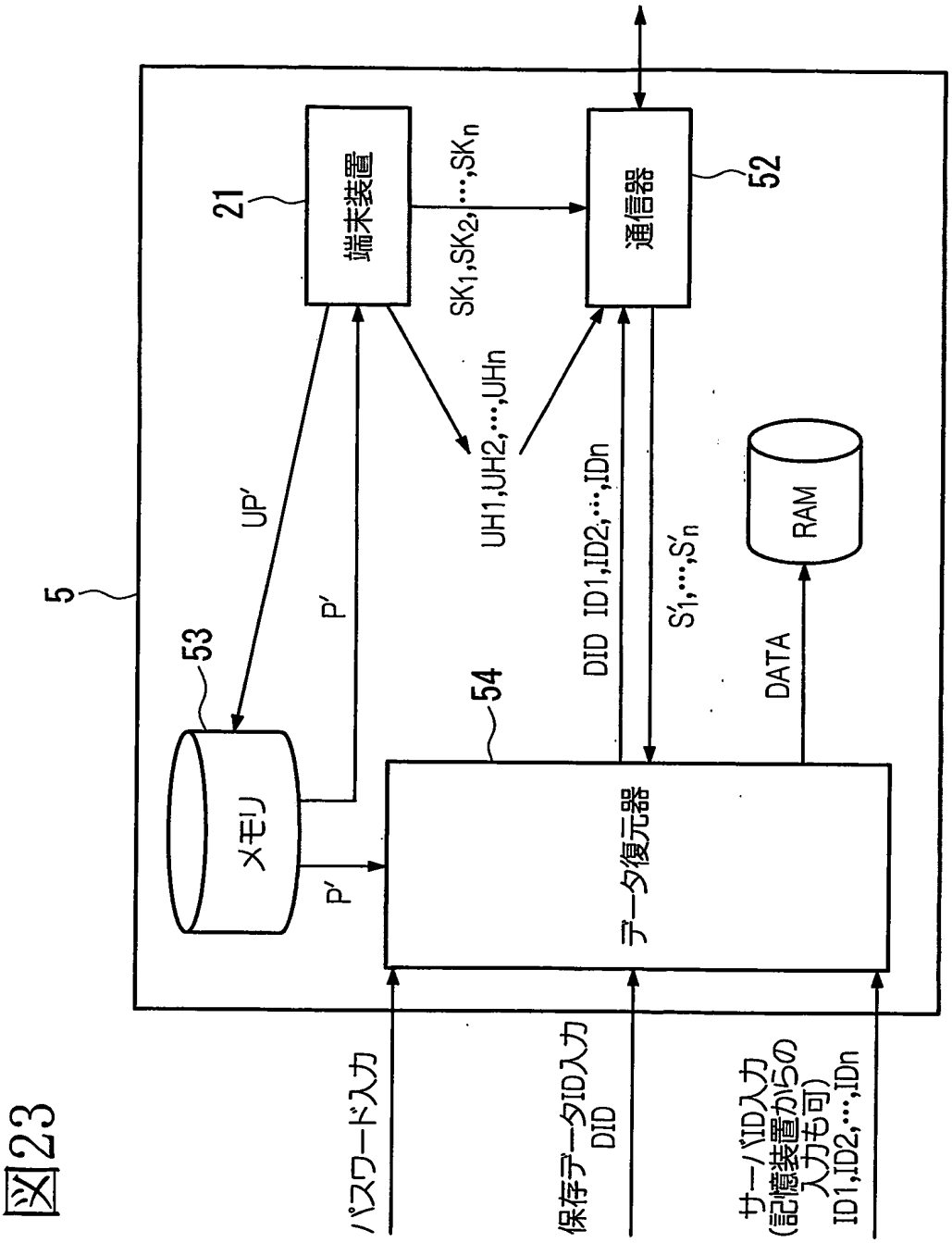
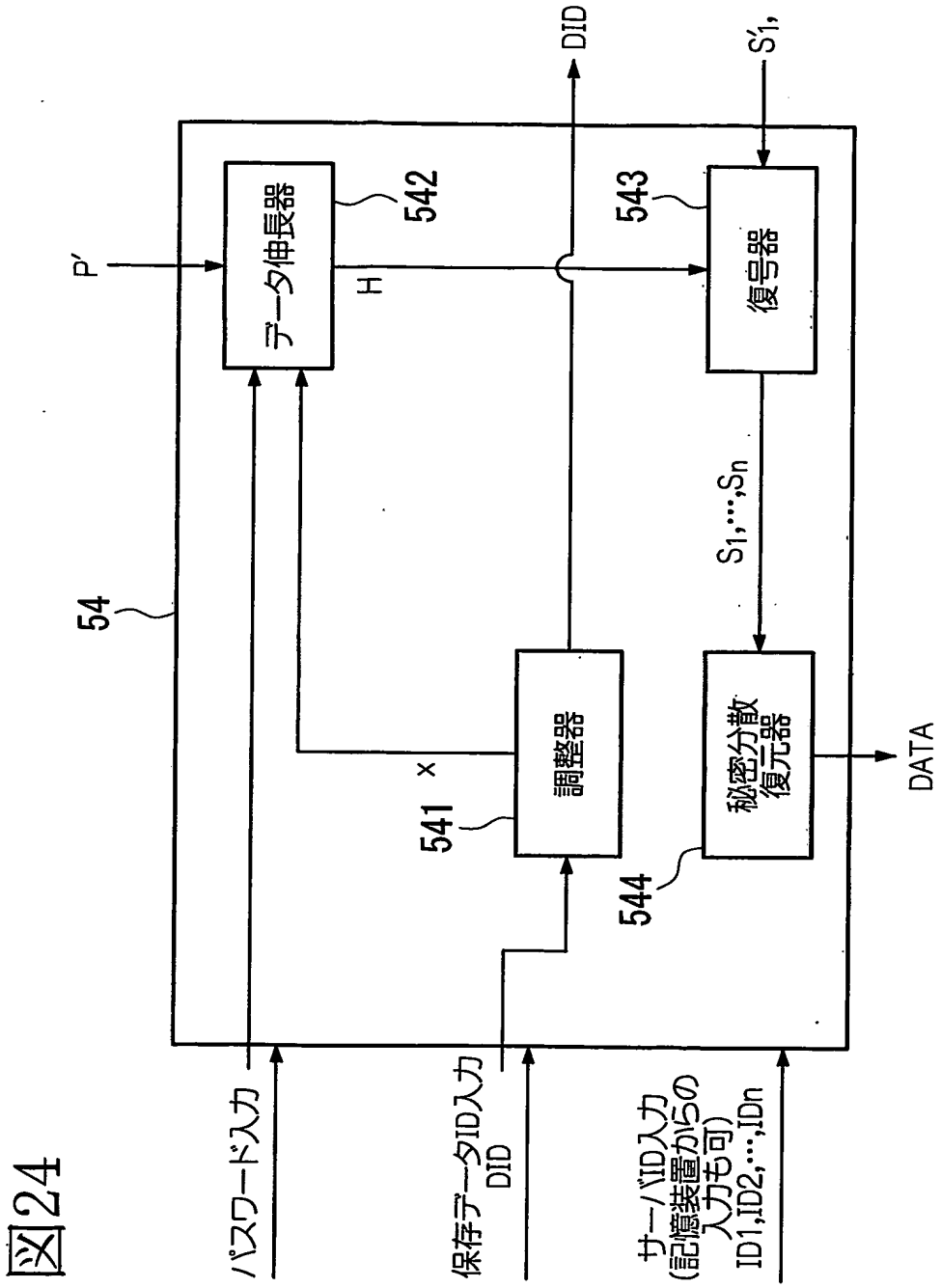

5
端末装置
21
通信器
52
SK1,SK2,…,SKn
UH1,UH2,…,UHn
UP′
P′
53
メモリ
54
データ復元器
DID ID1,ID2,…,IDn
S′1,…,S′n
DATA
RAM
パスワード入力
保存データID入力
DID
サーバID入力
(記憶装置からの
入力も可)
ID1,ID2,…,IDn

図23

図24


54
P'
データ伸長器
542
H
DID
S'1,
543
復号器
S1,…,Sn
x
541
調整器
544
秘密分散
復元器
DATA
パスワード入力
保存データID入力
DID
サーバID入力
(記憶装置からの
入力も可)
ID1,ID2,…,IDn

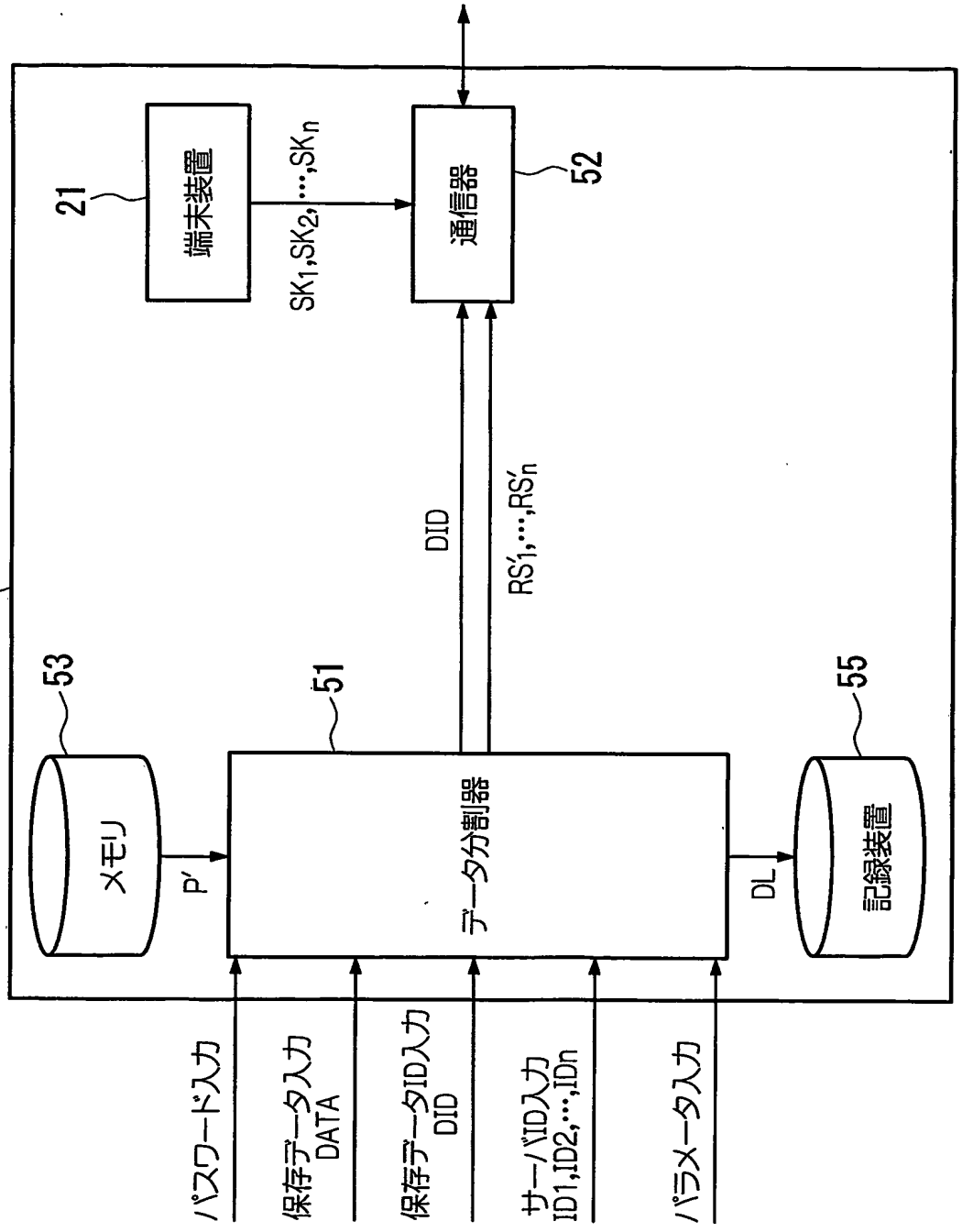
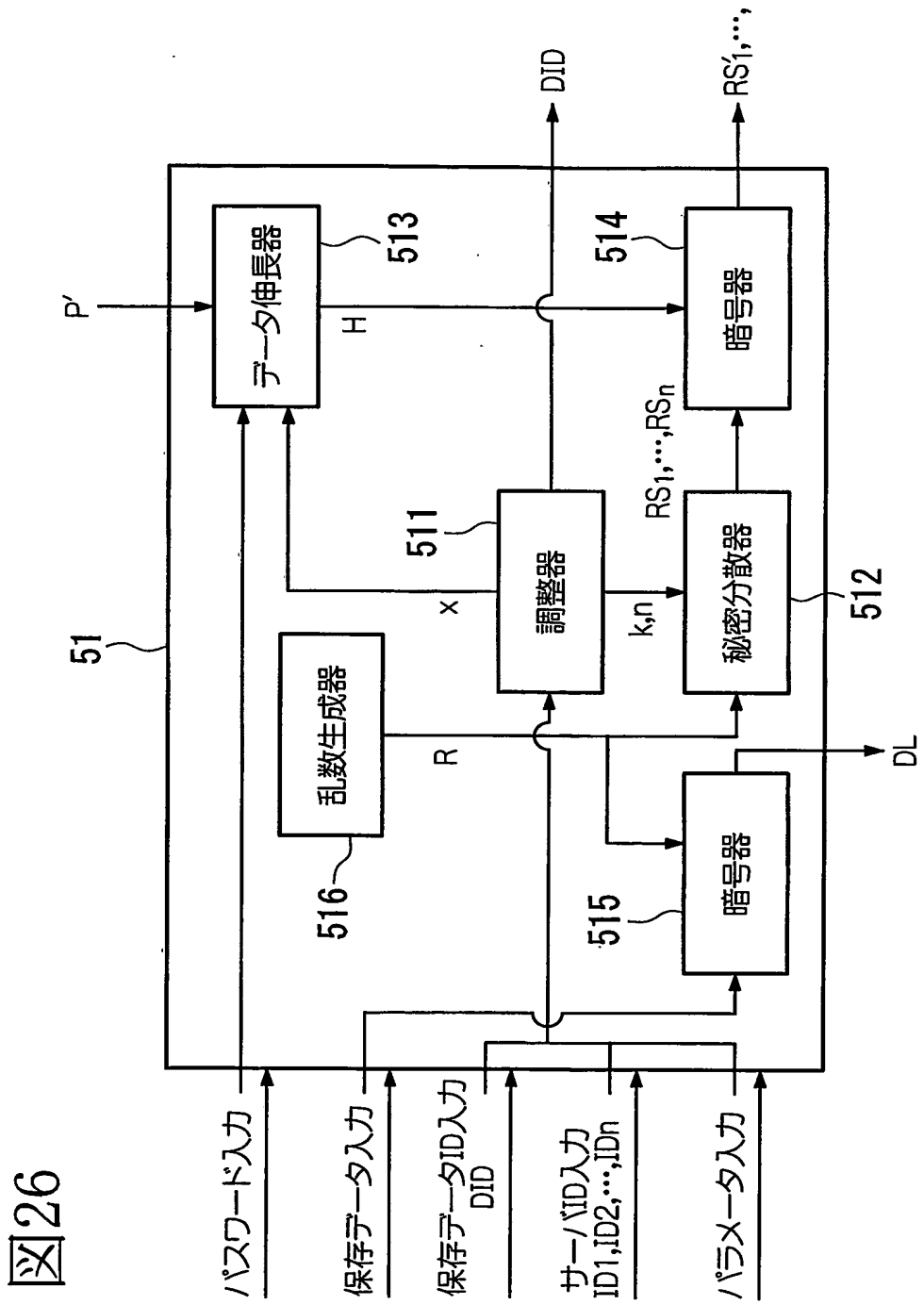

5
21
端末装置
SK1,SK2,…,SKn
通信器
52
DID
RS'1,…,RS'n
53
51
55
メモリ
P'
データ分割器
DL
記録装置
パスワード入力
保存データ入力 DATA
保存データID入力 DID
サーバID入力 ID1,ID2,…,IDn
パラメータ入力

図25

図26


51
P'
データ伸長器
513
H
DID
RS1',…,
514
暗号器
RS1,…,RSn
511
調整器
x
k,n
秘密分散器
512
乱数生成器
516
R
515
暗号器
DL
パスワード入力
保存データ入力
保存データID入力
DID
サーバID入力
ID1,ID2,…,IDn
パラメータ入力

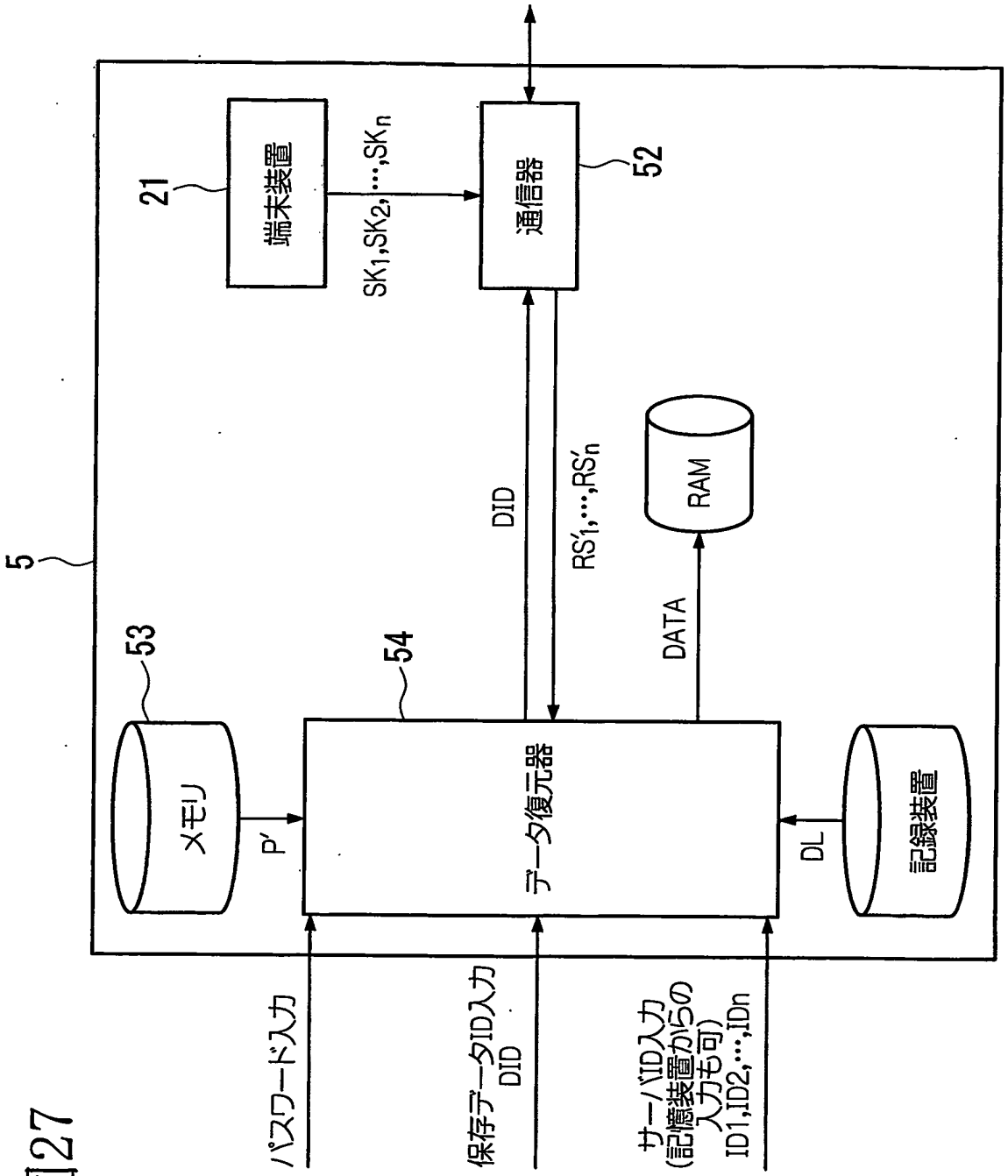

5
21
端末装置
SK1,SK2,…,SKn
通信器
52
DID
RS'1,…,RS'n
RAM
DATA
53
54
メモリ
P'
データ復元器
DL
記録装置
パスワード入力
保存データID入力
DID
サーバID入力
(記憶装置からの
入力も可)
ID1,ID2,…,IDn

図27

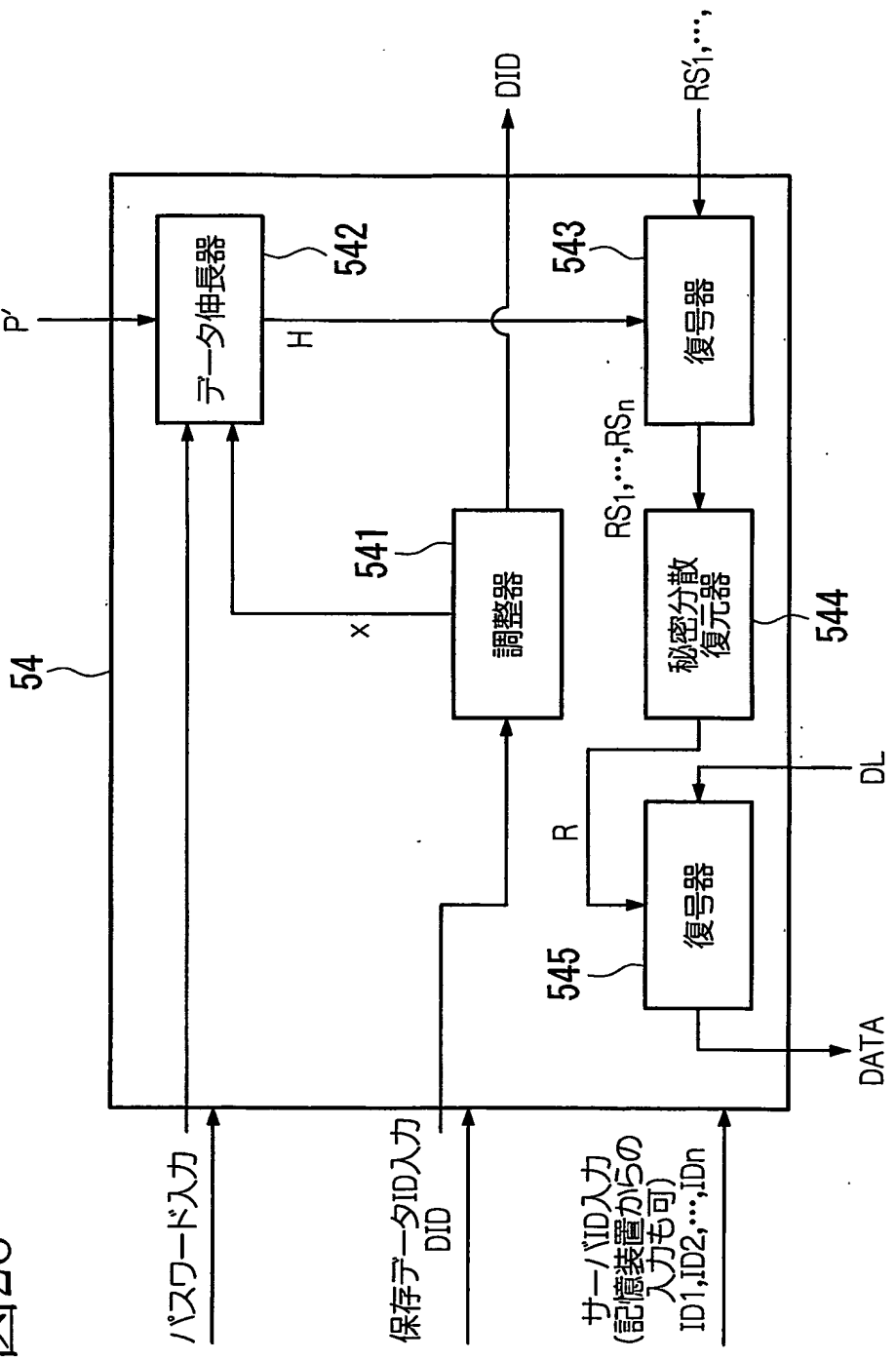

DID
RS1,…,
P'
データ伸長器
542
543
復号器
H
RS1,…,RSn
541
調整器
秘密分散
復元器
544
x
54
DL
R
復号器
545
DATA
パスワード入力
保存データID入力
DID
サーバID入力
(記憶装置からの
入力も可)
ID1,ID2,…,IDn

図28

INTERNATIONAL SEARCH REPORT

International application No.
PCT/JP2004/015184

A. CLASSIFICATION OF SUBJECT MATTER
Int.Cl[7] H04L9/32, H04L9/08, G06F15/00

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
Int.Cl[7] H04L9/32, H04L9/08, G06F15/00

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Jitsuyo Shinan Koho	1922-1996	Toroku Jitsuyo Shinan Koho	1994-2004
Kokai Jitsuyo Shinan Koho	1971-2004	Jitsuyo Shinan Toroku Koho	1996-2004

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
JICST FILE(JOIS)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X Y	Kazukuni KOBARA and Hideki IMAI, "Pretty-Simple Password-Authenticated Key-Exchange Protocol Proven to be Secure in the Standard Model", IEICE TRANSACTIONS on Fundamentals of Electronics, Communications and Computer Sciences, Vol.E85-A, No.10, 01 October, 2002 (01.10.02), pages 2229 to 2237 pages 2229 to 2237	1-46 47-50
Y	JP 4-245287 A (Matsushita Electric Industrial Co., Ltd.), 01 September, 1992 (01.09.92), Par. No. [0005]; Fig. 1 (Family: none)	47-50

☒ Further documents are listed in the continuation of Box C. ☐ See patent family annex.

* Special categories of cited documents:
"A" document defining the general state of the art which is not considered to be of particular relevance
"E" earlier application or patent but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed
"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
10 December, 2004 (10.12.04)	28 December, 2004 (28.12.04)
Name and mailing address of the ISA/ Japanese Patent Office Facsimile No.	Authorized officer Telephone No.

Form PCT/ISA/210 (second sheet) (January 2004)

INTERNATIONAL SEARCH REPORT

International application No.
PCT/JP2004/015184

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
P,A	SeongHan Shin, Kazukuni KOBARA, Hideki IMAI, "A New Password-based Authentication Protocol", Computer Security Symposium 2003 (CSS2003), Information Processing Society of Japan, Vol.2003, No.15, 29 October, 2003 (29.10.03), pages 7 to 12	1-50

国際調査報告	国際出願番号　ＰＣＴ／ＪＰ２００４／０１５１８４

A.　発明の属する分野の分類（国際特許分類（ＩＰＣ））

Ｉｎｔ．Ｃｌ⁷　Ｈ０４Ｌ９／３２，Ｈ０４Ｌ９／０８，Ｇ０６Ｆ１５／００

B.　調査を行った分野

調査を行った最小限資料（国際特許分類（ＩＰＣ））

Ｉｎｔ．Ｃｌ⁷　Ｈ０４Ｌ９／３２，Ｈ０４Ｌ９／０８，Ｇ０６Ｆ１５／００

最小限資料以外の資料で調査を行った分野に含まれるもの

日本国実用新案公報	１９２２－１９９６年
日本国公開実用新案公報	１９７１－２００４年
日本国登録実用新案公報	１９９４－２００４年
日本国実用新案登録公報	１９９６－２００４年

国際調査で使用した電子データベース（データベースの名称、調査に使用した用語）

ＪＩＣＳＴファイル（ＪＯＩＳ）

C.　関連すると認められる文献

引用文献のカテゴリー＊	引用文献名　及び一部の箇所が関連するときは、その関連する箇所の表示	関連する請求の範囲の番号
X Y	Kazukuni KOBARA and Hideki IMAI, “Pretty-Simple Password-Authenticated Key-Exchange Protocol Proven to be Secure in the Standard Model”, IEICE TRANSACTIONS on Fundamentals of Electronics, Communications and Computer Sciences, VOL. E85-A, NO. 10, 2002. 10. 01, p. 2229-2237 p. 2229-2237	１－４６ ４７－５０

☒ Ｃ欄の続きにも文献が列挙されている。　　☐ パテントファミリーに関する別紙を参照。

＊　引用文献のカテゴリー

「Ａ」特に関連のある文献ではなく、一般的技術水準を示すもの

「Ｅ」国際出願日前の出願または特許であるが、国際出願日以後に公表されたもの

「Ｌ」優先権主張に疑義を提起する文献又は他の文献の発行日若しくは他の特別な理由を確立するために引用する文献（理由を付す）

「Ｏ」口頭による開示、使用、展示等に言及する文献

「Ｐ」国際出願日前で、かつ優先権の主張の基礎となる出願の日の後に公表された文献

「Ｔ」国際出願日又は優先日後に公表された文献であって出願と矛盾するものではなく、発明の原理又は理論の理解のために引用するもの

「Ｘ」特に関連のある文献であって、当該文献のみで発明の新規性又は進歩性がないと考えられるもの

「Ｙ」特に関連のある文献であって、当該文献と他の１以上の文献との、当業者にとって自明である組合せによって進歩性がないと考えられるもの

「＆」同一パテントファミリー文献

国際調査を完了した日	国際調査報告の発送日
１０．１２．２００４	28.12.2004

国際調査機関の名称及びあて先	特許庁審査官（権限のある職員）		
日本国特許庁（ＩＳＡ／ＪＰ） 郵便番号１００－８９１５ 東京都千代田区霞が関三丁目４番３号	青木　重徳	5M	4229
	電話番号　０３－３５８１－１１０１　内線　３５９７		

Ｃ（続き）．　関連すると認められる文献

引用文献のカテゴリー＊	引用文献名　及び一部の箇所が関連するときは、その関連する箇所の表示	関連する請求の範囲の番号
Ｙ	ＪＰ　４－２４５２８７　Ａ（松下電器産業株式会社）　１９９２．０９．０１，第【０００５】段落，図１　（ファミリーなし）	４７－５０
Ｐ，Ａ	SeongHan Shin, Kazukuni Kobara, Hideki Imai, “A New Password-based Authentication Protocol”，コンピュータセキュリティシンポジウム２００３（ＣＳＳ２００３），社団法人情報処理学会，Ｖｏｌ．２００３，Ｎｏ．１５，２００３．１０．２９，ｐ．７－１２	１－５０